HEAT TRANSFER IN INFINITE SLABS

by

Donald Ray McNeill

United States
Naval Postgraduate School

THESIS

HEAT TRANSFER IN INFINITE SLABS

by

Donald Ray McNeill

December 1970

Heat Transfer in Infinite Slabs

by

Donald Ray McNeill
Lieutenant Commander, United States Navy
B. S., United States Naval Academy, 1962

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1970

ABSTRACT

An analysis is made of termperature distributions in a slab of finite thickness and infinite extent, one surface of which is perfectly insulated and the other surface of which is exposed to a fluid the temperature of which varies in a specified way. Two cases are considered. In the first case, the temperature of the fluid is suddenly changed and maintained at the new value. In the second case the temperature of the fluid increases linearly with time. From the solutions of these problems, curve sheets are developed which permit evaluating each of eight different quantities of engineering significance. These curve sheets cover more cases of interest and are easier to use than similar curves which have appeared in the report literature.

TABLE OF CONTENTS

TABLE OF SYMBOLS AND ABBREVIATIONS

a		plate thickness (ft.)
C_n	=	$4 \sin M_n/(2M_n + \sin 2M_n)$ (dimensionless)
E		rate of linear temperature change (deg.F/hour)
F		magnitude of step temperature change (deg.F)
h		convective heat transfer coefficient ($BTU/hr\text{-}ft^2\text{-}deg.F$)
k		thermal conductivity (BTU/hr-ft-deg.F)
M_n		n_{th} root of the equation $M \tan M = N_B$ (dimensionless)
N_B	=	Biot number = ha/k (dimensionless)
T		temperature (deg.F)
t		time (hours)
x		distance into slab measured from insulated surface (ft.)
α		thermal diffusivity (ft^2/hr)
β_n	=	M_n/a (ft^{-1})
λ_n	=	$M_n\sqrt{\alpha}/a = \beta_n\sqrt{\alpha}$ ($hr^{-\frac{1}{2}}$)
Θ	=	Fourier number $= \alpha t/a^2$ (dimensionless)
ρ_n	=	$M_n^2\Theta = \lambda_n^2 t$ (dimensionless)

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Professor John E. Brock for his guidance and encouragement during the course of the investigation. His personal notes were used extensively in the derivation of all equations presented in the thesis in addition to footnoted references.

I. INTRODUCTION

The problem of convective heat transfer by a fluid flowing through a thin-wall pipe with a well insulated exterior surface is of considerable importance, particularly in connection with assessing the effects of rapid operating transients in nuclear power systems. Several years ago contractors engaged in this sort of work began using a set of several sheets of curves which facilitate these calculations. The writer has been unable to learn the original source of these curves or even to find them anywhere in the open literature. However, they are to be found in such documents as References 2 and 3.

The theory upon which these curves are based is well known and will be developed, for reference purposes, in Section II of this thesis. It replaces the thin-wall tube by a flat plate which extends infinitely in the y and z directions and is of finite thickness a in the x direction. At the surface $x = 0$ it is presumed that there is a perfect insulating layer and at the surface $x = a$ the plate is exposed to a fluid whose temperature varies with time in a specified manner; the coefficient of surface heat transfer between the fluid and the plate is a specified constant h. The initial temperature distribution in the plate is given as a function of x. The problem is to determine the subsequent temperature distribution in the plate as a function of x and the time t. It is further presumed that the appropriate physical properties of the plate are constants.

This problem can be reduced to that of solving a certain linear partial differential equation subject to certain initial and boundary conditions. To simplify and systematize the analysis it is convenient to introduce and

deal with several dimensionless ratios. There is more than one form in which the solution may be represented but we will obtain and develop a particular and convenient form.

In the case of a pipe the wall thickness of which is an appreciable fraction of its radius, cylindrical geometry should be employed and an additional geometric parameter, ratio of wall thickness to radius, must be introduced. The solution in this case is more complicated than that which will be developed here and which forms the basis for the useful curves referred to above. However, in most practical cases the wall thickness is indeed sufficiently small compared to radius to justify using the flat-plate geometry which will be employed here. No attempt is made in this thesis to explore the differences between results using flat-plate geometry and results using cylindrical geometry.

Two particular cases arise frequently in practice. In each case it is assumed that the fluid and the material of the pipe are at the same temperature prior to the time $t = 0$. In one case, referred to as "step", it is assumed that the fluid <u>suddenly</u> assumes and maintains a new and different temperature starting at time $t = 0$. In the other case, referred to as "ramp" or "linear", it is assumed that starting at time $t = 0$ the temperature of the fluid changes linearly with time. By superposition of results, the cases of piecewise linear changes in fluid temperature including sudden "jumps" in temperature can be accommodated, if convenient solutions are available for the two cases described.

The useful curves referred to above give significant information relating to these two cases. However, those sets of curves which have come to hand are considerably less useful than would be desirable because they do not provide a sufficiently broad range of significant parameter

values to represent physical situations actually encountered and they require some rather drastic interpolation and extrapolation. Furthermore, certain integral properties of the solutions are useful for certain types of stress calculation and the curve sets previously available have not provided these properties. It should also be noted that some of the information contained in the curves which are developed could also be obtained from the well known "Heisler charts," specifically from Figures 7 and 10 of Reference 4.

Accordingly, it has been a major purpose of this thesis to reconstruct the curve sets, including a greater variation in significant parameters and to augment the collection of curves by adding several new curves likely to prove useful in engineering practice. The curves developed were checked and verified to be correct by comparison to previously published curves. The only exceptions to this were the ΔT_1 and ΔT_2 cases for which there were no comparison curves or data available.

The development, including the differential equation and boundary and initial conditions, leading to the production of the desired curves is presented in Section II.A. of this thesis. In Section II.B. the theory of the step fluid temperature change is developed and that of the linear fluid temperature change is developed in Section II.C. A discussion of the computational method used to obtain the curve sets is presented in Section II.D. Additionally, the computer program has many comments inserted to aid in understanding it. A brief discussion of how to use the curve sets is presented in the first part of Appendix A, following which the curves themselves are presented. Further instruction is given in References 2 and 3. The person who is engaged in heat transfer calculations involving slabs or thin wall pipes can easily develop further methods as required.

Inasmuch as it is likely that in nuclear power plant calculations there may be need to perform calculations relating to convective heat transfer of the type described above on a massive scale, involving hundreds of different parameter sets and hundreds of transient specifications, it is of interest to consider ways in which human intervention is not required to "read" curves. Of course, a computer may be used to calculate each of these possibly hundreds of thousands of cases using the exact theory developed herein. However, since the calculations are, as will be seen, quite demanding, there is reason to seek an approximation which will have sufficient accuracy for most purposes but which will provide significant speed-up of the calculations. For this purpose, in Section III hereof, we seek to determine the coefficients of a quickly-evaluated approximation in such a manner that the error will be limited to small values over the entire significant ranges of all parameters.

Figure 1 illustrates the problem which will be dealt with.

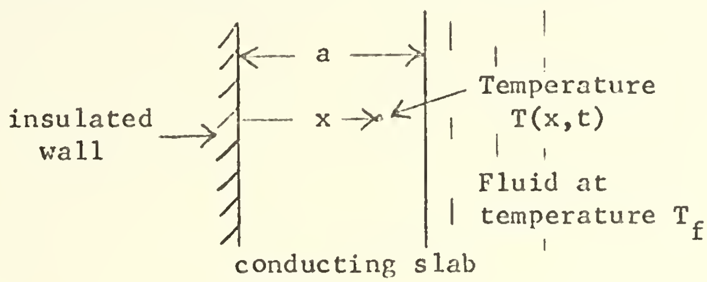


Figure 1

We will find a convenient dimensionless temperature $T(x,t)/T_f$ for the two previously mentioned types of fluid temperature change. The step temperature change, where $T_f = 0$ for time $t < 0$ and $T_f = F =$ const. for $t > 0$, will be considered first in Section II.B. The linear temperature change, where $T_f = 0$ for $t < 0$ and $T_f = Et$ for $t > 0$ will then be considered in Section II.C. Here $E =$ const. is in degrees per unit time. Actual solutions will be cast in the dimensionless form $T(x,t)/T_f$ which will

be an explicit function of x or x/a and Fourier number and an implicit function of Biot number.

Each of the two general equations giving $T = T(x,t)$ will then be employed to provide eight particular properties most likely to be of use to the designer. Curves representing these properties in a convenient way will be obtained and exhibited in Appendix A. These results include those which have appeared in the curve sets of References 2 and 3 but with a fuller coverage of significant parameter values. Additionally some new properties are presented that are intended to assist in performing evaluations according to the rules of the ANSI B31.7 Code for Nuclear Power Piping and a forthcoming edition of Section III, Nuclear Pressure Vessels, of the ASME Boiler and Pressure Vessel Code.

II. THEORY AND DEVELOPMENT OF EQUATIONS FOR HEAT TRANSFER

A. GENERAL THEORY

It is desired to find T(x,t) such that it satisfies the unsteady heat conduction equation in one dimension, namely

$$\alpha \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t} \qquad (1)$$

The initial and boundary conditions of the problem are:

1) $T(x,0) = 0$ (slab initially at uniform temperature throughout, and we take this to be zero without loss of generality)

2) $\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0$ (perfect insulation at x = 0)

3) $\left.\frac{\partial T}{\partial x}\right|_{x=a} = \frac{h}{k}\left[T_f - T(a,t)\right]$ (continuity of heat transfer at fluid-solid interface)

B. STEP FLUID TEMPERATURE CHANGE

Here $T_f = F = \text{const.}$ for $t > 0$. The solution to this problem is found[1] to be

$$\frac{T(x,t)}{F} = 1 - \sum C_n \cos(\beta_n x)\, e^{-\rho_n} \qquad (2)$$

where:

a) $\rho_n = M_n^2 \Theta = \lambda_n^2 t$

b) $\lambda_n = M_n \sqrt{\alpha}/a$

c) $\beta_n = M_n/a$

---

[1]Schneider, Paul J., Conduction Heat Transfer, p. 250, Addison Wesley, 1955.

d) $C_n = 4 \sin M_n / (2M_n + \sin 2M_n)$

e) Σ denotes summation over $n = 1,2,3,\ldots\infty$

f) The numbers M_n are an infinite number of distinct roots as seen in Figure 2 below, of the equation

$$M \tan M = N_B \tag{3}$$

where N_B is the Biot number

$$N_B = ha/k$$

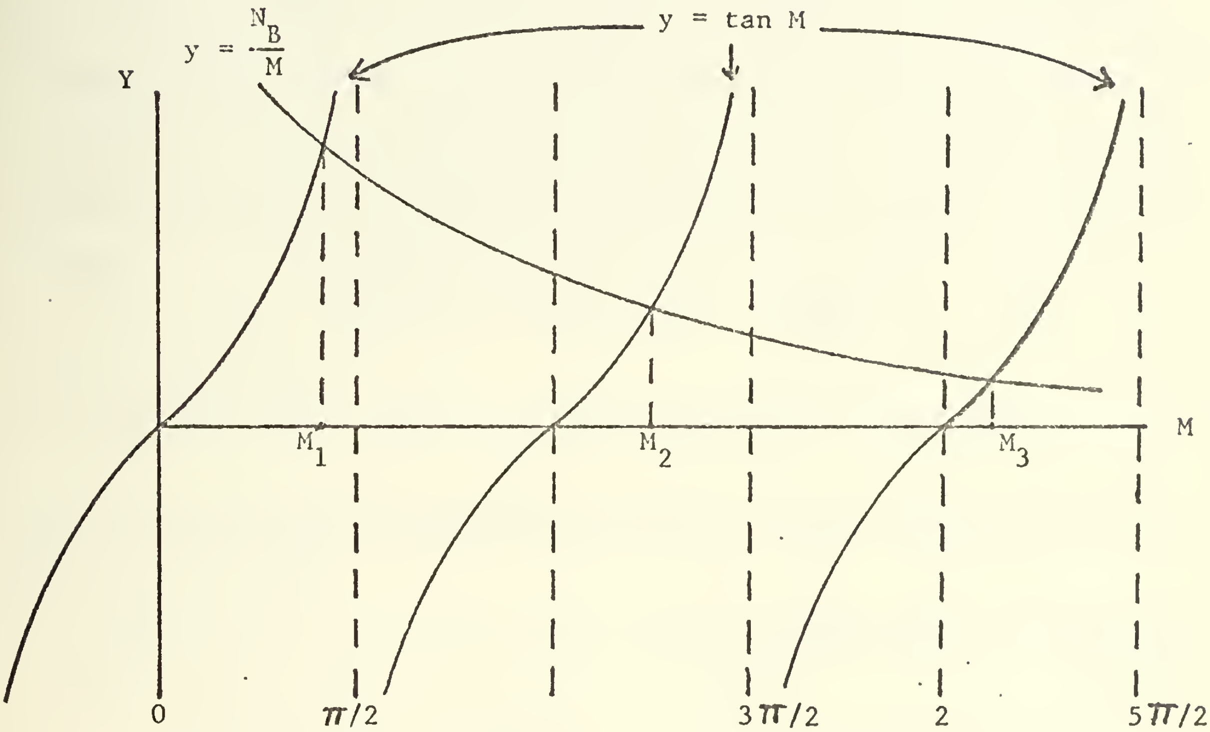


Figure 2

To verify that the governing differential equation is satisfied we find

$$\frac{\partial T}{\partial x} = F \Sigma C_n \beta_n \sin(\beta_n x)\, e^{-\lambda_n^2 \tau} \tag{4}$$

$$\frac{\partial^2 T}{\partial x^2} = F \Sigma C_n \beta_n^2 \cos(\beta_n x)\, e^{-\lambda_n^2 \tau}$$

$$\frac{\partial T}{\partial \tau} = F \Sigma C_n \lambda_n^2 \cos(\beta_n x)\, e^{-\lambda_n^2 \tau}$$

The governing equation is then

$$\alpha F \sum C_n \beta_n^2 \cos(\beta_n x) e^{-\lambda_n^2 t} = F \sum C_n \lambda_n^2 \cos(\beta_n x) e^{-\lambda_n^2 t}$$

But $\alpha \beta_n^2 = \alpha M_n^2/a^2 = \lambda_n^2$. Therefore the differential equation is satisfied. Next, we consider the first boundary condition which takes the form

$$1 = \sum C_n \cos(\beta_n x)$$

and we will show that this is true by actually determining the quantities C_n. These are similar to Fourier coefficients which must represent the constant value of unity. Even though the argument $(\beta_n x)$ is not simply a multiple of n, the customary technique of evaluation can still be used. Multiply through by $\cos(\beta_m x)dx$ and integrate from 0 to a. On the left we get

$$\int_0^a \cos(\beta_m x)\,dx = \frac{\sin M_m}{\beta_m}$$

On the right we get

$$\sum C_n \int_0^a \cos(\beta_m x) \cos(\beta_n x)\,dx$$

and it is necessary to distinguish between the cases where m = n and m $\neq$ n.

For m $\neq$ n, we find

$$\int_0^a \cos(\beta_m x) \cos(\beta_n x)\,dx = \frac{a}{2}\left[\frac{\sin(M_m - M_n)}{M_m - M_n} + \frac{\sin(M_m + M_n)}{M_m + M_n}\right]$$

$$= a\left[\frac{\cos M_m \cos M_n}{M_m^2 - M_n^2}\right]\left[M_m \tan M_m - M_n \tan M_n\right]$$

which vanishes since M_m and M_n are distinct roots of $M \tan M = N_B$.

For m = n, we find

$$\int_0^a \cos^2(\beta_n x)\,dx = \frac{1}{\beta_n}\int_0^{M_n} \cos^2\phi\,d\phi = \frac{a}{2}\left[1 + \frac{\sin 2M_n}{2M_n}\right]$$

Thus, we have established that

$$\frac{a}{M_n} \sin M_n = \frac{a}{2} C_n \left[1 + \frac{\sin 2M_n}{2M_n} \right]$$

so that $C_n = 4 \sin M_n / (2M_n + \sin 2M_n)$

as previously defined.

That the second boundary condition is satisfied may be immediately verified by inspection of equation (4). Additionally, upon noting that

$$\beta_n \sin(\beta_n a) = \frac{M_n}{a} \sin M_n = \frac{h}{K} \cos M_n$$

it is trivial to show that the third boundary condition is satisfied. Thus the governing equation and boundary conditions have been shown to be satisfied. Equation (2) is interesting and useful in its own right. However, we now proceed to derive formulas for eight particularly useful consequences of the relation. In these formulas the distance x no longer appears as a parameter and the dependence upon time is incorporated in the value of the Fourier number

$$\Theta = \text{Fourier number} = \alpha t / a^2$$

which is simply related to the exponent in the general formula, viz:

$$\rho_n = M_n^2 \Theta$$

The eight cases we consider and the corresponding formulas are:

1. Case 1. The average temperature of the slab is

$$\frac{T_{av}(t)}{F} = \frac{1}{a} \int_0^a \frac{T(x,t)}{F} dx = \frac{1}{a} \left| x - \sum_1 \frac{C_n \sin(\beta_n x) e^{-\rho_n}}{\beta_n} \right|_0^a$$

$$= \frac{1}{a} \left[a - \sum_1 \frac{C_n \sin M_n e^{-\rho_n}}{\beta_n} \right] = 1 - \sum_1 \frac{C_n \sin M_n e^{-\rho_n}}{M_n}$$

2. Case 2. The heated surface temperature is, by inspection,

$$\frac{T(a,t)}{F} = 1 - \sum C_n \cos M_n \, e^{-P_n}$$

3. Case 3. The insulated surface temperature is, by inspection,

$$\frac{T(0,t)}{F} = 1 - \sum C_n e^{-P_n}$$

4. Case 4. The difference between heated surface and average temperature is, by comparing Cases 1 and 2,

$$\frac{T(a,t) - T_{av}(t)}{F} = \sum C_n e^{-P_n} \left(\frac{\sin M_n}{M_n} - \cos M_n \right)$$

5. Case 5. The difference between average and insulated surface temperature is, by comparing Cases 1 and 3,

$$\frac{T_{av}(t) - T(0,t)}{F} = \sum C_n e^{-P_n} \left(1 - \frac{\sin M_n}{M_n} \right)$$

6. Case 6. The difference between heated and insulated surface temperature is, by comparing Cases 2 and 3,

$$\frac{T(a,t) - T(0,t)}{F} = \sum C_n e^{-P_n} \left(1 - \cos M_n \right)$$

7. Case 7. The "Equivalent Linear Variation" is denoted by the symbol ΔT_1. This quantity, which is defined by equation (5), is useful in determining what are known as secondary stresses due to radial thermal gradients in piping.

$$\frac{\Delta T_1}{F} = \frac{12}{a^2} \int_0^a \left(x - \frac{a}{2} \right) \frac{T(x,t)}{F} \, dx \tag{5}$$

$$= \frac{12}{a^2} \int_0^a \left\{ x - \frac{a}{2} - \sum C_n e^{-P_n} \left[x \cos(\beta_n x) - \frac{a}{2} \cos(\beta_n x) \right] \right\} dx$$

$$= \sum \frac{C_n e^{-P_n}}{M_n^2} \left[12 \left(1 - \cos M_n \right) - 6 M_n \sin M_n \right]$$

8. Case 8. The "Peak Component of Thermal Gradient" is denoted by the symbol ΔT_2. This quantity, which is defined by equation (6), is useful in determining what are called peak stresses due to the nonlinearity of the radial thermal gradient in piping.

$\Delta T_2/F =$

$$\text{Max}\left\{ \left|T(a,\tau)-T_{av}(\tau)\right|-\tfrac{1}{2}\left|\Delta T_1\right|,\ \left|T_{av}(\tau)-T(0,\tau)\right|-\tfrac{1}{2}\left|\Delta T_1\right|,\ 0 \right\}/F \tag{6}$$

It should be remarked that the definitions of ΔT_1 and ΔT_2 given above are those which at the time of this writing have obtained the approval of the Working Group on Piping of the Design Subcommittee on Section III of the ASME Boiler and Pressure Vessel Committee. It is probable that they will appear in a forthcoming edition of the Code, but several stages of approval remain before this can happen.

C. LINEAR FLUID TEMPERATURE CHANGE

Here $T_f = Et$ for $t > 0$. The solution is obtained[1] by Duhamel's method which gives

$$T(x,\tau) = E\left[\tau - \sum \frac{C_n \cos(\beta_n x)\left(1-e^{-\lambda_n^2 \tau}\right)}{\alpha \beta_n^2}\right] \tag{7}$$

or

$$T(x,\tau) = E\tau\left[1 - \sum \frac{C_n \cos(\beta_n x)\left(1-e^{-\lambda_n^2 \tau}\right)}{\lambda_n^2 \tau}\right] \tag{8}$$

since $\lambda_n^2 = \alpha \beta_n^2$.

To show that this solution satisfies the differential equation, we get from equation (8):

$$\frac{\partial T}{\partial x} = E\tau\left[\sum \frac{\beta_n C_n \sin(\beta_n x)\left(1-e^{-\lambda_n^2 \tau}\right)}{\lambda_n^2 \tau}\right] = E \sum \frac{C_n \sin(\beta_n x)\left(1-e^{-\lambda_n^2 \tau}\right)}{\lambda_n^2/\beta_n}$$

[1] Personal notes of Professor John E. Brock

$$\frac{\partial^2 T}{\partial x^2} = E \sum \frac{C_n \cos(\beta_n x)(1-e^{-\lambda_n^2 \tau})}{\lambda_n^2/\beta_n^2} = E \sum \frac{C_n \cos(\beta_n x)(1-e^{-\lambda_n^2 \tau})}{\alpha}$$

and from equation (7):

$$\frac{\partial T}{\partial \tau} = E\left[1 - \sum C_n \cos(\beta_n x) e^{-\lambda_n^2 \tau}\right]$$

However, the constants C_n were previously determined so as to satisfy the equation $1 = \sum C_n \cos(\beta_n x)$.

Thus we can write $\frac{\partial T}{\partial \tau} = E \sum C_n \cos(\beta_n x)(1-e^{-\lambda_n^2 \tau})$

and it is clear that we have satisfied the differential equation

$$\alpha \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial \tau}$$

Now consider the boundary conditions. From equation (7) it is clear that $T(x,0) = 0$ and from our expression for $\frac{\partial T}{\partial x}$ it is clear that $\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0$.

Finally, considering the third boundary condition, we have

$$\left.\frac{\partial T}{\partial x}\right|_{x=a} = E \sum \frac{C_n \sin M_n (1-e^{-\lambda_n^2 \tau})}{\alpha \beta_n}$$

and finding T(a,t) from equation (7) gives

$$\frac{h}{k}\left[T_f - T(a,\tau)\right] = E \frac{h}{k} \sum C_n \frac{\cos M_n (1-e^{-\lambda_n^2 \tau})}{\alpha \beta_n^2}$$

$$= E \sum \frac{C_n \sin M_n (1-e^{-\lambda_n^2 \tau})}{\alpha \beta_n}\left[\frac{h \cos M_n}{k \beta_n \sin M_n}\right]$$

However, the last term in parenthesis is recognized as

$$N_B/M \tan M = 1$$

so that the third boundary condition is also satisfied. The solution may now be expressed in the more desirable form

$$\frac{T(x,\tau)}{E\tau} = 1 - \sum \frac{C_n \cos(\beta_n x)(1-e^{-\rho_n})}{\rho_n} \tag{8}$$

The eight special cases for the linear problem are, using the same notation as before:

1. <u>Case 1</u>.

$$\frac{T_{av}(t)}{Et} = \frac{1}{a}\int_0^a \frac{T(x,t)}{Et}\,dx = \frac{1}{a}\left| x - \sum \frac{c_n \sin(\beta_n x)(1-e^{-\rho_n})}{\beta_n \rho_n} \right|_0^a$$

$$= 1 - \sum \frac{c_n \sin M_n (1-e^{-\rho_n})}{M_n \rho_n}$$

2. Case 2. By inspection

$$\frac{T(a,t)}{Et} = 1 - \sum \frac{c_n \cos M_n (1-e^{-\rho_n})}{\rho_n}$$

3. Case 3. By inspection

$$\frac{T(0,t)}{Et} = 1 - \sum \frac{c_n (1-e^{-\rho_n})}{\rho_n}$$

4. Case 4. From Cases 1 and 2

$$\frac{T(a,t) - T_{av}(t)}{Et} = \sum \frac{c_n (1-e^{-\rho_n})}{\rho_n} \left(\frac{\sin M_n}{M_n} - \cos M_n \right)$$

5. Case 5. From Cases 1 and 3

$$\frac{T_{av}(t) - T(0,t)}{Et} = \sum \frac{c_n (1-e^{-\rho_n})}{\rho_n} \left(1 - \frac{\sin M_n}{M_n} \right)$$

6. Case 6. From Cases 2 and 3

$$\frac{T(a,t) - T(0,t)}{Et} = \sum \frac{c_n (1-e^{-\rho_n})}{\rho_n} \left(1 - \cos M_n \right)$$

7. Case 7. Again using the proposed definition

$$\frac{\Delta T_1}{Et} = \frac{12}{a^2}\int_0^a \left(x - \frac{a}{2}\right) \frac{T(x,t)}{Et}\,dx$$

$$= \sum \frac{c_n (1-e^{-\rho_n})}{\rho_n} \left[12(1-\cos M_n) - 6 M_n \sin M_n \right]$$

8. Case 8. Again

$$\frac{\Delta T_2}{Et} = \text{Max} \left\{ \left| T(a,t) - T_{av}(t) \right| - \tfrac{1}{2}\left|\Delta T_1\right|,\ \left| T_{av}(t) - T(0,t) \right| - \tfrac{1}{2}\left|\Delta T_1\right|,\ 0 \right\} / Et$$

D. COMPUTATIONAL SCHEME

The curves, which are exhibited in Appendix A, were obtained by computation on the IBM 360/67 computer at the Naval Postgraduate School. The programs which were used are presented in Appendix B. There should be no

difficulty in following these programs, which were written in FORTRAN IV language using double precision arithmetic; they are quite straightforward implementations of the equations developed in the preceding sections of this thesis.

In covering the range of parameters involved, values of the Biot number were introduced in a dimensioned array while the Fourier numbers were generated by a DO-loop. A large number (41) of values of Fourier numbers were used; these are the abscissa values for the curves. A smaller number of values of Biot number were employed; these correspond to the individual curves on each sheet. It was necessary to select these values by trial and error in each case to permit a reasonably attractive distribution of curves on each sheet.

In calculating the roots of the equation $M \tan M = N_B$, Newton's method was used; a root was declared to be established when the next approximation made less difference than 10^{-8}. The number of roots actually determined and the number of terms in the summations was thirty. No specific investigation was made of the effect of taking a larger or smaller number of terms; however, some earlier work done by Professor Brock indicates that sufficient accuracy for graph plotting could probably have been obtained with only a few terms.

A remark is called for with regard to the plotting routine. Although the calculations were performed with double precision arithmetic, the plotting routine, a local library subroutine called DRAW, accepts only single precision inputs. Therefore, as a final step prior to plotting, the appropriate arrays had to be changed from double to single precision format.

As previously noted, the program should be easy to follow with the comments that have been inserted. If it is desired to change the number

and/or value of the Biot numbers for any particular problem, the value of "KELL" for that problem and the particular BN(J) array would have to be changed. The width and/or interval spacing of the abscissae ($\log_{10}\Theta$) can be changed by changing "KALL" (the number of points plotted for each Biot number) or "FANG" (the interval spacing for the abscissae). The starting point for this interval is set by the constant A in the card

```
LOGTH(I) = A + (I-1)*FANG
```

which appears in each particular problem. Many "WRITE" statements were included to provide a printout of data in addition to the graphs.

III. DEVELOPMENT OF APPROXIMATE EQUATIONS

A. THEORY

As can be seen from the foregoing developments, the calculation of temperature at any given time in the interior of a slab which is convectively heated in the ways we have considered is a rather demanding task. Of course modern digital computers can perform such evaluations rapidly, but there are a large number of arithmetic operations involved in each such computation. Automated analysis of complete piping systems subjected to a variety of thermal transient conditions might require hundreds of thousands of such evaluations and in such a case it would be valuable to have a sufficiently accurate approximation which could be evaluated in a fraction of the time required for an evaluation using the theoretically exact equations.

Thus we seek to find an accurate representation of T(x,t) in the form T = w(x,y,z) where x,y and z are conveniently chosen functions of x/a, N_B, and Θ respectively* and the function w is to be determined. We shall attempt to find the functions x, y and z so as to permit covering the values of x/a and the many decades of N_B and of Θ that we wish to cover and still provide fairly uniform accuracy over the entire range. For the function w we can think of no more useful representation than a power series

$$w = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{p=1}^{P} A_{mnp}\, x^{m-1} y^{n-1} z^{p-1} \qquad (1)$$

where the coefficients A_{mnp} are to be determined from a large number of point quadruples (w_i, x_i, y_i, z_i) obtained by use of the exact solution. A

*Note that x which appears hereafter is a specified function of the parameter (x/a). It should not be confused with the distance x which was used previously.

customary way of doing this is to determine the A_{mnp} so as to minimize the sum of the squares of the errors $(w_i - w(x_i, y_i, z_i))$, where w is the approximate solution and w_i the known value from the exact solution. We now proceed to establish the theory of determining the A_{mnp} to do this. Let the positive integers M, N, P , and Q be given where Q is the number of sets of four numbers x_i, y_i, z_i, and w_i $(i = 1, 2, \ldots Q)$. Determine the A_{mnp} in equation (1) so that

$$E = \sum_{i=1}^{Q} (w - w_i)^2$$

is a minimum. The minimizing condition is

$$0 = \frac{1}{2}\frac{\partial E}{\partial A_{rst}} = \sum_{i=1}^{Q}\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{p=1}^{P}\left[A_{mnp}\, x_i^{m-1} y_i^{n-1} z_i^{p-1} - w_i\right]\left[x_i^{r-1} y_i^{s-1} z_i^{t-1}\right]$$

where $r = 1,2,\ldots M$; $s = 1,2,\ldots N$; $t = 1,2,\ldots P$

or

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{p=1}^{P} A_{mnp} \sum_{i=1}^{Q} x_i^{m+r-2}\, y_i^{n+s-2}\, z_i^{p+t-2} = \sum_{i=1}^{Q} w_i\, x_i^{r-1}\, y_i^{s-1}\, z_i^{t-1} \tag{2}$$

We now introduce the notation

$$\beta_{JKL} = \sum_{i=1}^{Q} x_i^{J-1}\, y_i^{K-1}\, z_i^{L-1}$$

(Do not confuse this with an earlier and different use of the letter β).

with $J = m + r-1$; $K = n + s-1$; $L = p + t-1$

and $\gamma_{rst} = \sum_{i=1}^{Q} w_i\, x_i^{r-1}\, y_i^{s-1}\, z_i^{t-1}$

Equation (2) becomes

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{p=1}^{P} A_{mnp}\, \beta_{JKL} = \gamma_{rst} \tag{3}$$

Equation (3) is linear in the unknown coefficients A_{mnp} so that no essential difficulty is involved in solving for them. If M, N, and P are small integers (say no larger than 4) a brute force type of arrangement and rearrangement of the equations will provide a form which can easily be solved by computer to yield the desired quantities. However, if the M, N, and P are larger integers, some sublety may be required in devising a computer program to perform the necessary manipulations.

The problem of finding the correct values of the coefficients A_{mnp} becomes one of:

a) Selecting the three appropriate functions x, y, and z to be used.

b) Selecting values of M, N, and P. Larger values should give more accurate results at the expense of more computing time and larger storage requirements.

c) Generating values of β_{JKL} and γ_{rst} on the computer.

d) Solving for the A_{mnp} from equation (3) by linear methods.

e) Checking the approximate solution against the exact solution.

As a simple example of this process consider the case where M = N = P = 2. This gives the following system.

$$\begin{bmatrix}
\beta_{111} & \beta_{112} & \beta_{121} & \beta_{122} & \beta_{211} & \beta_{212} & \beta_{221} & \beta_{222} \\
 & \beta_{113} & \beta_{122} & \beta_{123} & \beta_{212} & \beta_{213} & \beta_{222} & \beta_{223} \\
 & & \beta_{131} & \beta_{132} & \beta_{221} & \beta_{222} & \beta_{231} & \beta_{232} \\
 & & & \beta_{133} & \beta_{222} & \beta_{223} & \beta_{232} & \beta_{233} \\
 & & & & \beta_{311} & \beta_{312} & \beta_{321} & \beta_{322} \\
 & \text{symmetric} & & & & \beta_{313} & \beta_{322} & \beta_{323} \\
 & & & & & & \beta_{331} & \beta_{332} \\
 & & & & & & & \beta_{333}
\end{bmatrix}
\begin{bmatrix} A_{111} \\ A_{112} \\ A_{121} \\ A_{122} \\ A_{211} \\ A_{212} \\ A_{221} \\ A_{222} \end{bmatrix}
=
\begin{bmatrix} \gamma_{111} \\ \gamma_{112} \\ \gamma_{121} \\ \gamma_{122} \\ \gamma_{211} \\ \gamma_{212} \\ \gamma_{221} \\ \gamma_{222} \end{bmatrix}$$

The A_{mnp} will provide an approximate solution over the range of data points used in the generation of β_{JKL} and γ_{rst}. We used the functions $x = \cos(x/a)$, $y = \Theta/(1+\Theta)$, and $z = N_B/(1 + N_B)$ and covered the ranges $0 \leq x/a \leq 1$, $.001 \leq \Theta \leq 100.$, and $.01 \leq N_B \leq 1000000.$

B. COMPUTATIONAL SCHEME

The first program, "Generation of Betas and Gammas," works as follows:

1. The Biot numbers and Fourier numbers needed to generate the data points are established as for the heat transfer curves.

2. 30 roots of $M \tan M = N_B$ are calculated.

3. The exact formula derived in Section II is used to generate 144 data points (i.e., Q = 144).

4. The values of β_{JKL} and γ_{rst} are then calculated using their defining equations.

The beta and gamma matrices are now assembled from their known elements and the A_{mnp} found[1] by solving the system

$$[\beta][A]=[\gamma]$$

for the unknown value of A_{mnp}. Then a program, hereinafter called "Check Program", uses the theory of Section III.A. to quickly generate the approximate solutions. Values were printed for x/a = 0 and x/a = 1 and compared to the results of the curves for x = 0 and x = a to verify the approximation. This program displays the values of the A_{mnp} for both the step and ramp cases and demonstrates how the approximate solution is used on the computer. The programs are shown in Appendix B and are self explanatory with the included comments.

---

[1]The SMIS program of Prof. Giles Cantin was used to do this.

However, the checks were so unsatisfactory that it must be concluded that a programming error remains undetected or that our choice of the functions x, y, and z was very unwise.

IV. RECOMMENDATIONS FOR FURTHER WORK

The approximate solutions generated by the "Check Program" in Appendix B were in such error that no specific recommendations can be made. It appears that either there was an error in the generation of the matrix elements and therefore in the evaluation of the A_{mnp} or that the selection of the functions x, y, and z was unwise. Time limitations do not permit investigating this further. An attempt to correct the approximation should include:

1. Review the "Generation of Betas and Gammas" program to insure that it in fact does properly generate the values of β_{JKL} and γ_{rst}.

2. Rewrite this same program so as to

   a) eliminate the argument INDEX from the arrays of the program so that the additional storage gained will allow any number of Q data points to be used. This will allow making the increments between data points smaller and extending the range, where possible, as follows:

   $$.001 \leq N_B \leq 1{,}000{,}000$$
   $$.0001 \leq \Theta \leq 1{,}000$$

   in order to make the approximation more accurate in the desired range of

   $$.01 \leq N_B \leq 1{,}000{,}000$$
   $$.001 \leq \Theta \leq 100$$

   b) employ double precision arithmetic.

   c) permit using larger values of M, N, and P.

3. Give more thought to the selection of appropriate functions of x/a, N_B and Θ which might improve the approximation without requiring larger values of M, N, and P.

4. Once the values of β_{JKL} and γ_{rst} are evaluated, insure that they are properly placed in their respective matrices and the coefficients A_{mnp} determined correctly.

5. Review the "Check Program" and insure that it correctly evaluates the approximate solution. All work should be done in double precision to insure the computer does not contribute to the error.

APPENDIX A

HEAT TRANSFER CURVES

Use of the curves is quite simple when the following procedure is applied:

1. Select the appropriate curve.

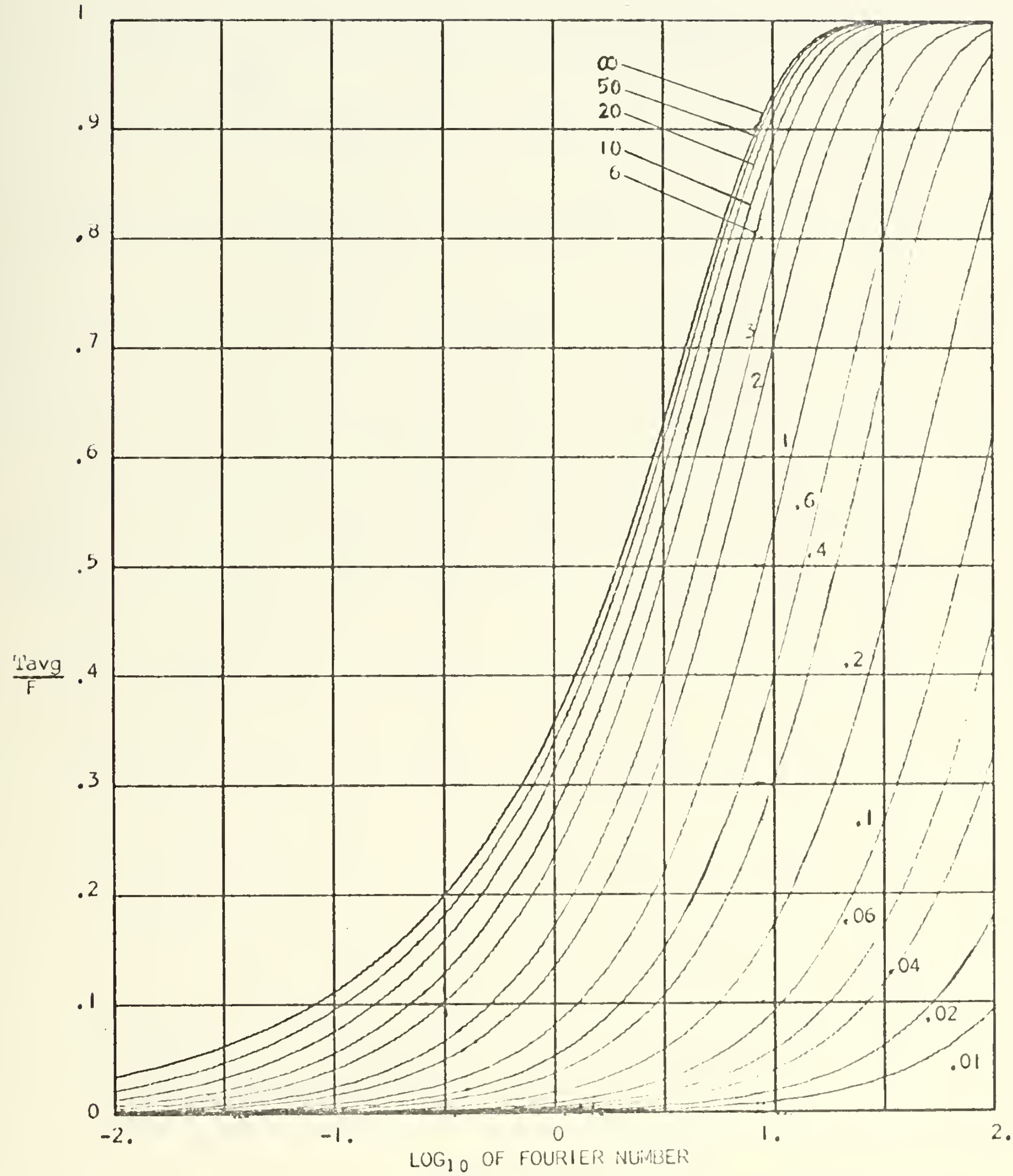
2. Determine the Biot number N_B = ha/k.

3. Determine the Fourier number $\Theta = \alpha t/a^2$ and $\log_{10} \Theta$.

4. Using $\log_{10} \Theta$ as abcissa, find the intersection with the appropriate curve corresponding to N_B (interpolate N_B as necessary).

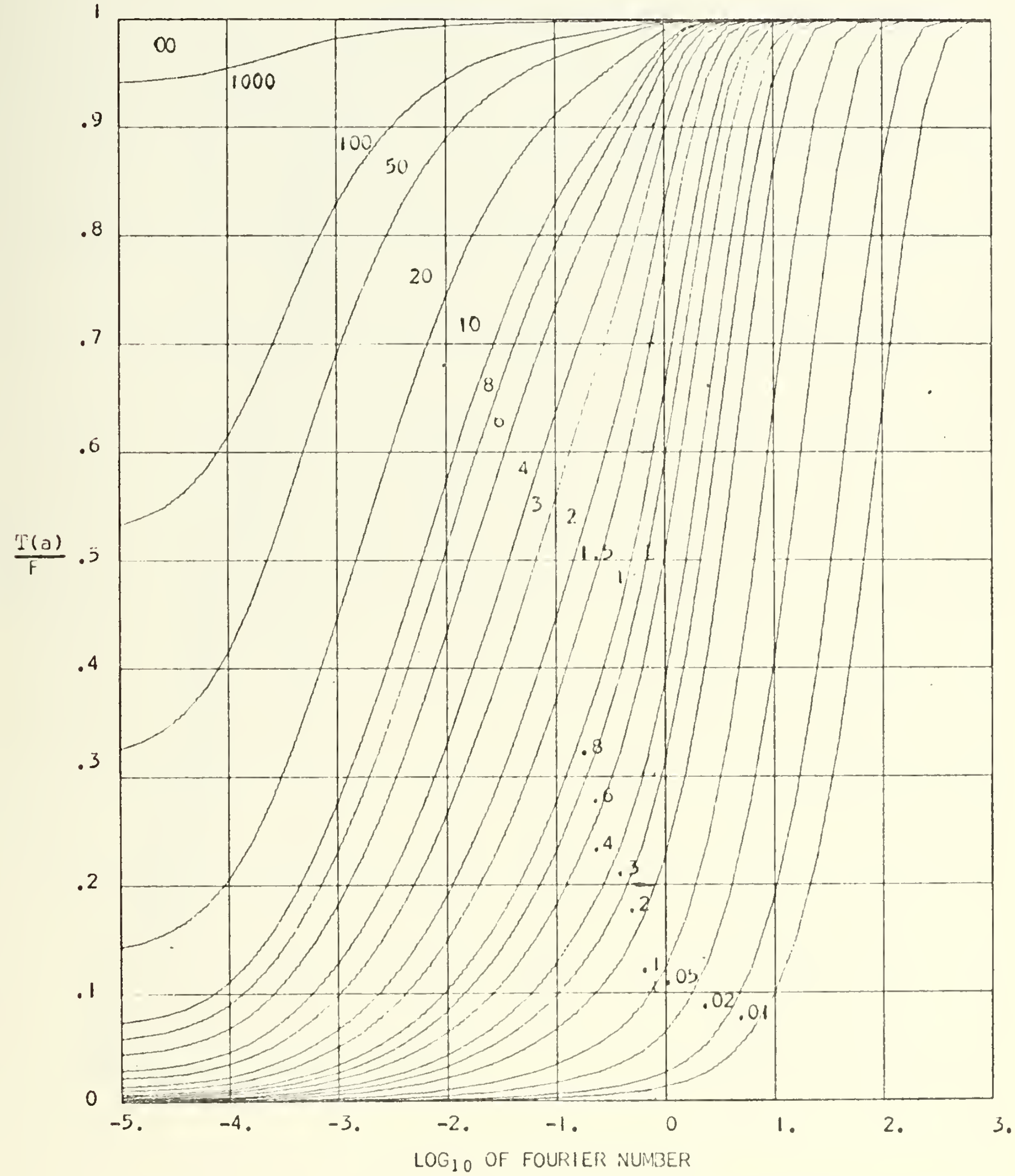
5. Go to the left at this height on the graph. On the axis of ordinates read the value of the desired result.

6. Perform any indicated arithmetic operation.



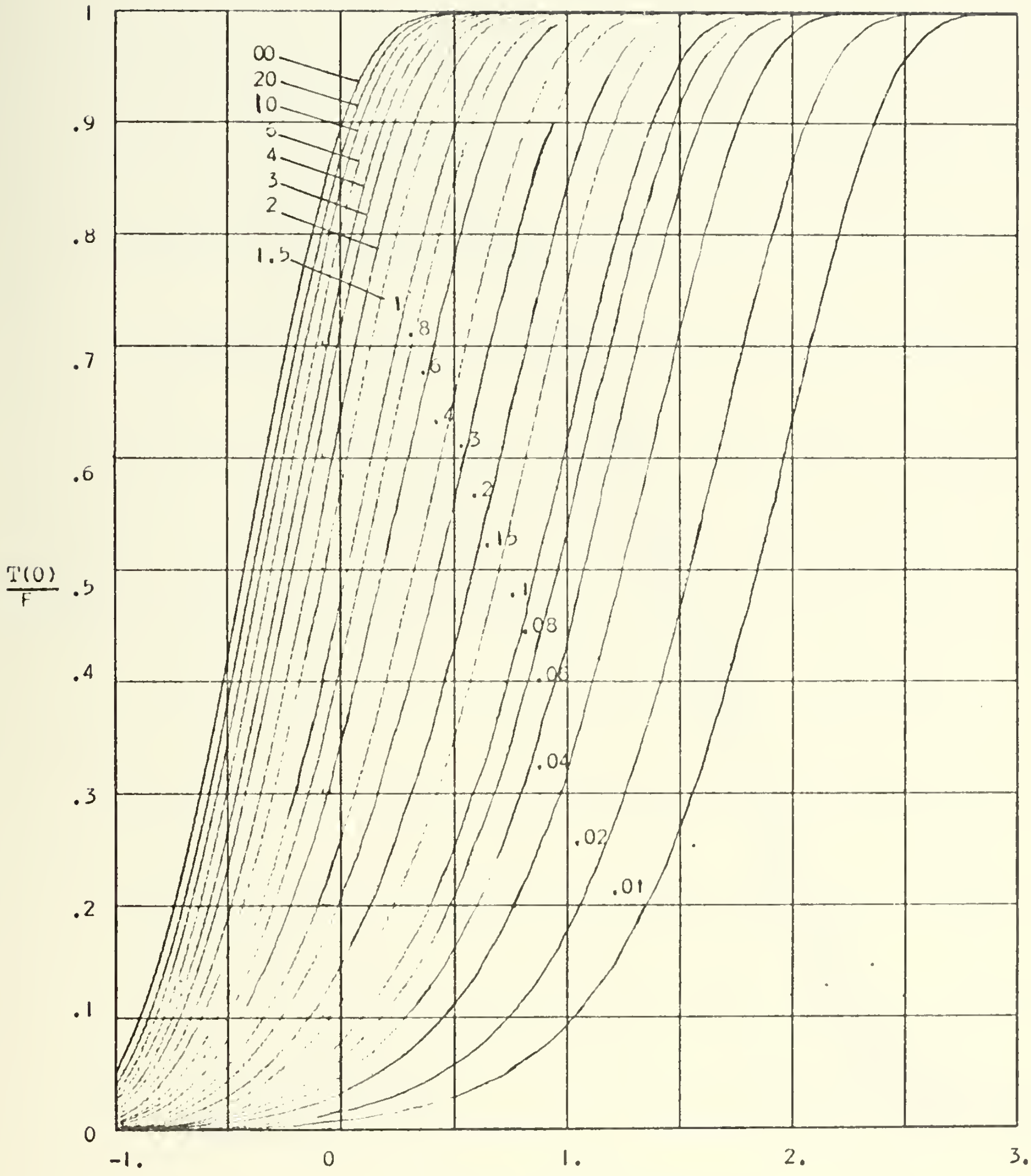
PLOT OF Tavg/F VS. LOG_{10} OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-1



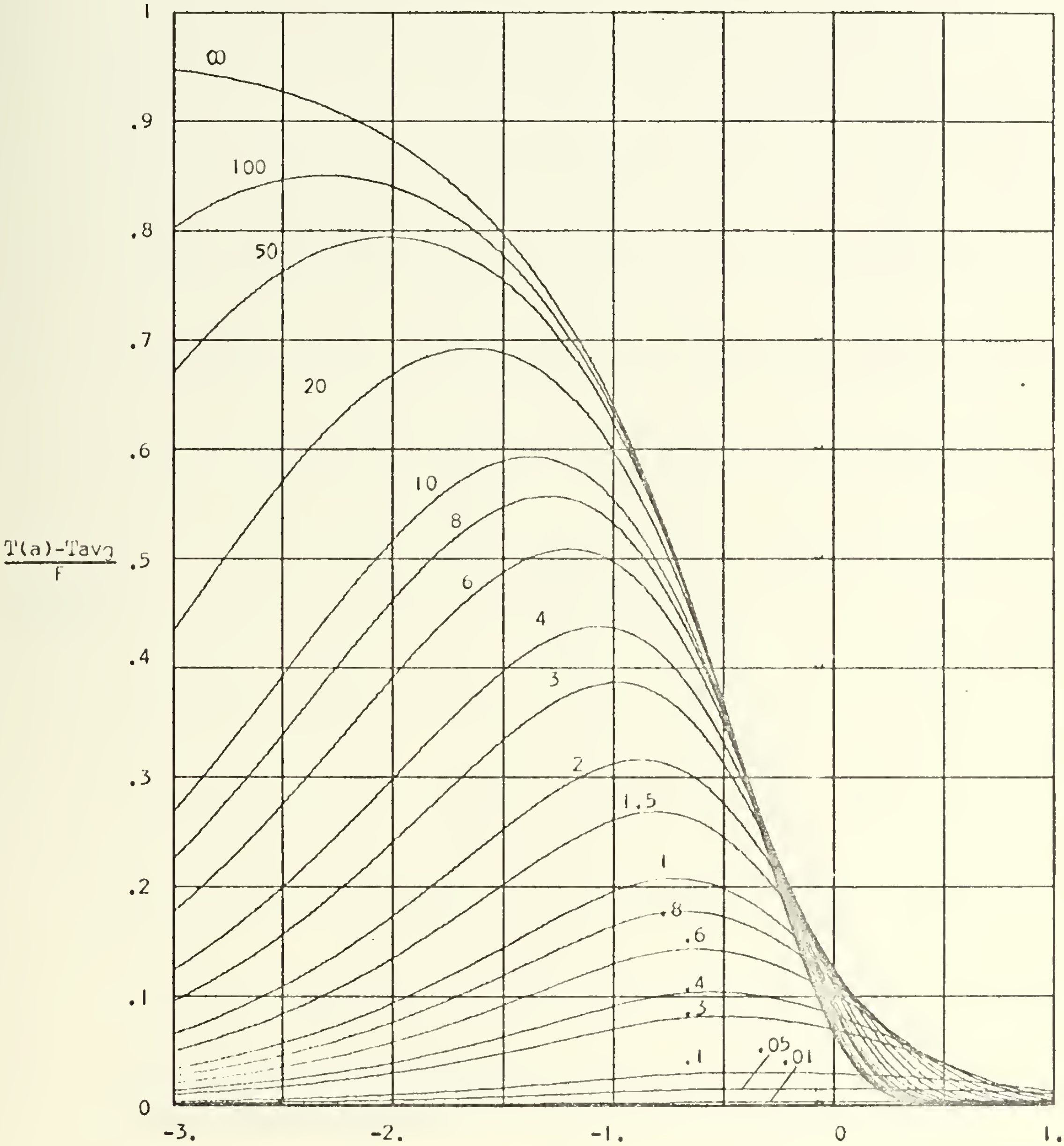
PLOT OF T(a)/F VS. LOG_{10} OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-2



PLOT OF T(0)/F VS. LOG_{10} OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

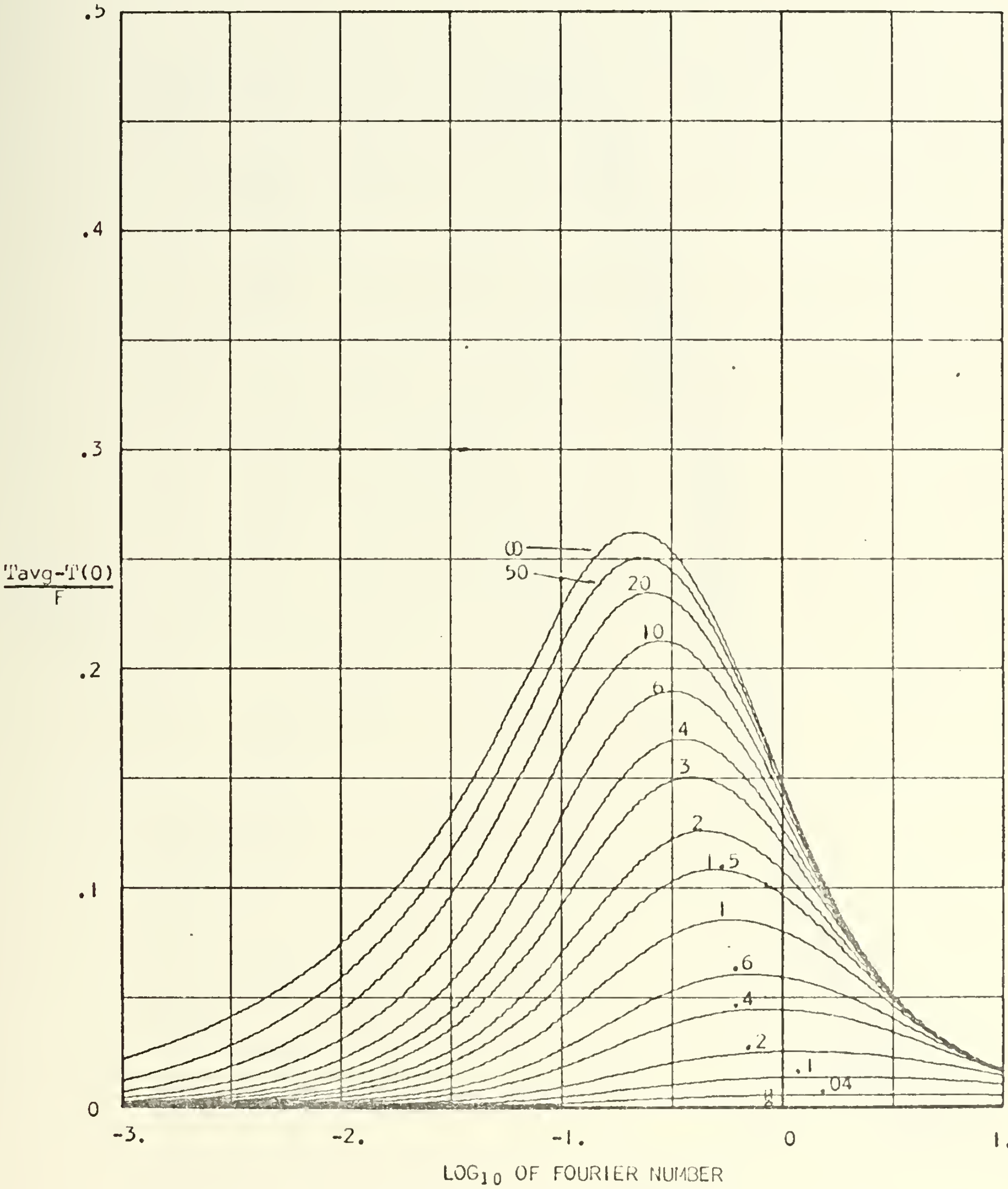
Fig. A-3



LOG$_{10}$ OF FOURIER NUMBER

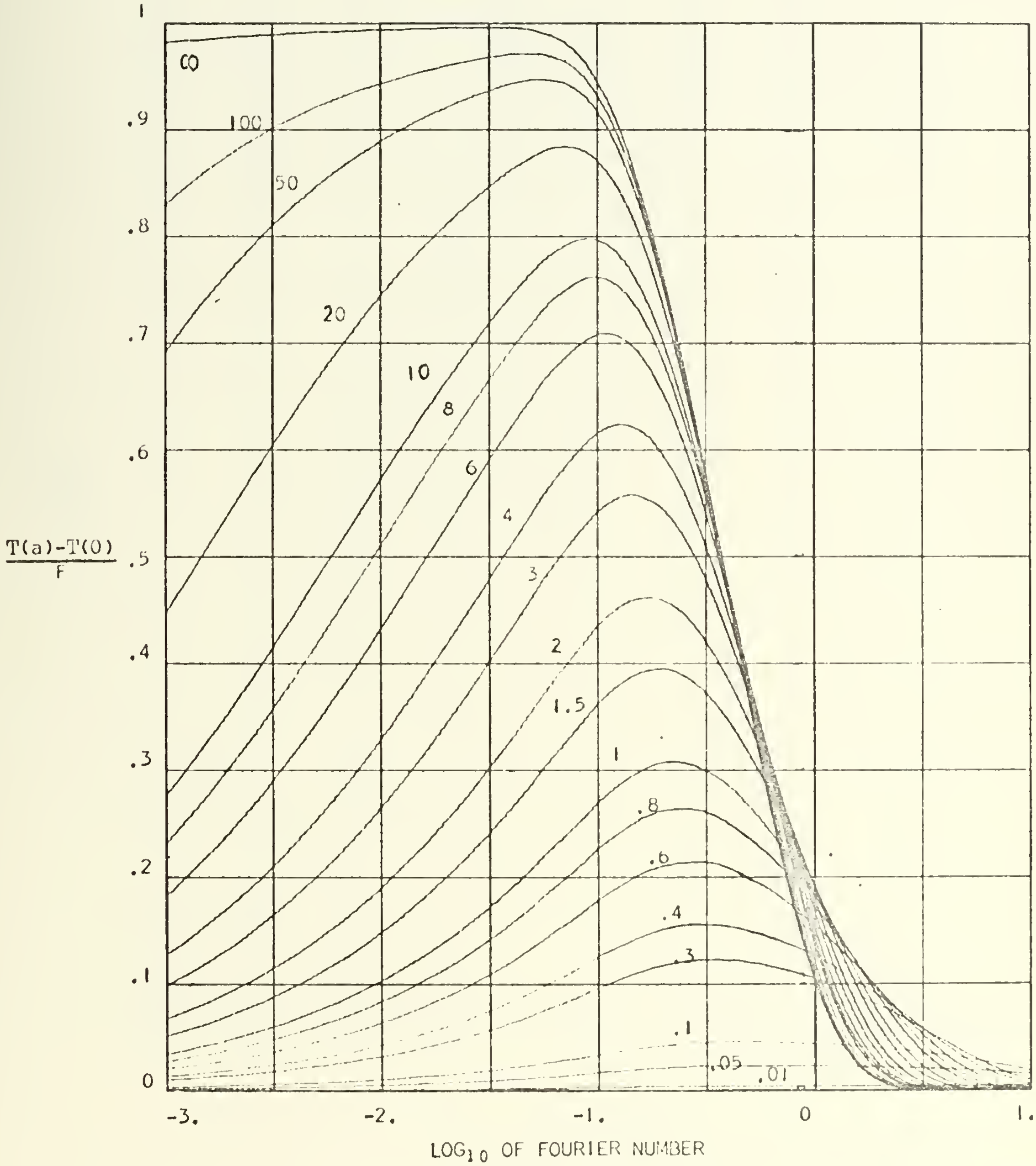
PLOT OF T(a)-Tavg/F VS. LOG$_{10}$ OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-4



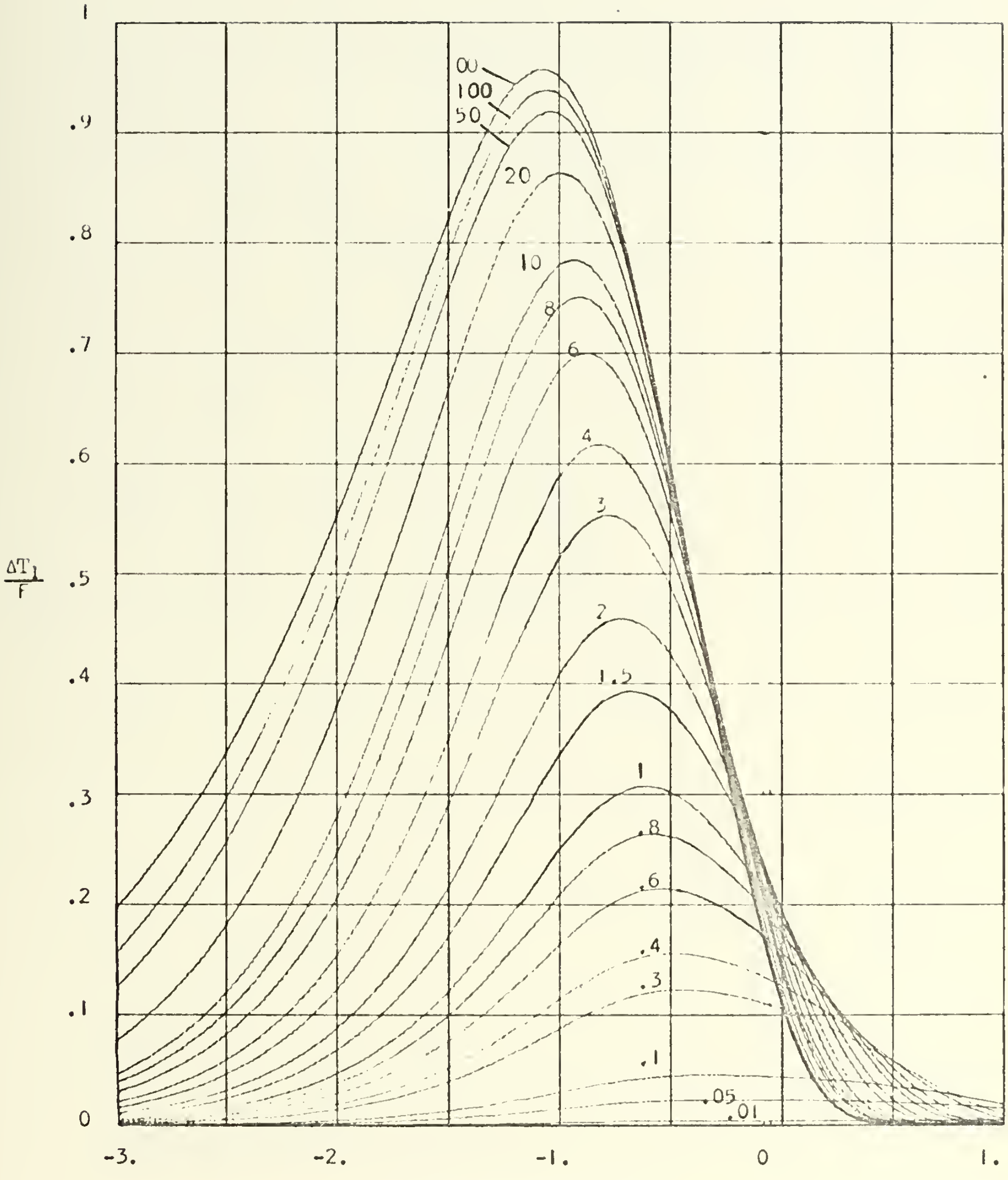
PLOT OF (Tavg - T(0))/F VS. LOG_{10} OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-5



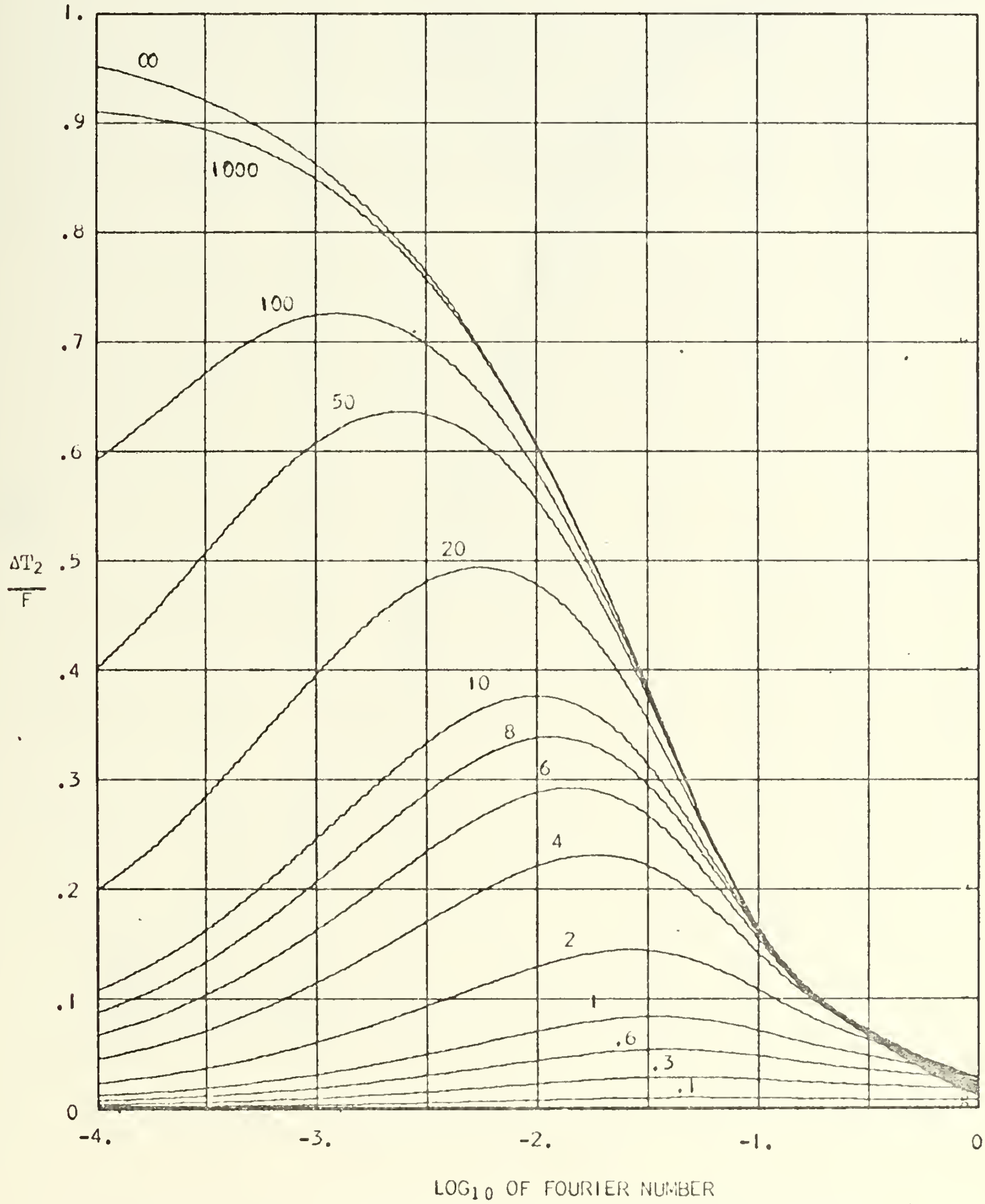
PLOT OF T(a)-T(0)/F VS. LOG$_{10}$ OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-6



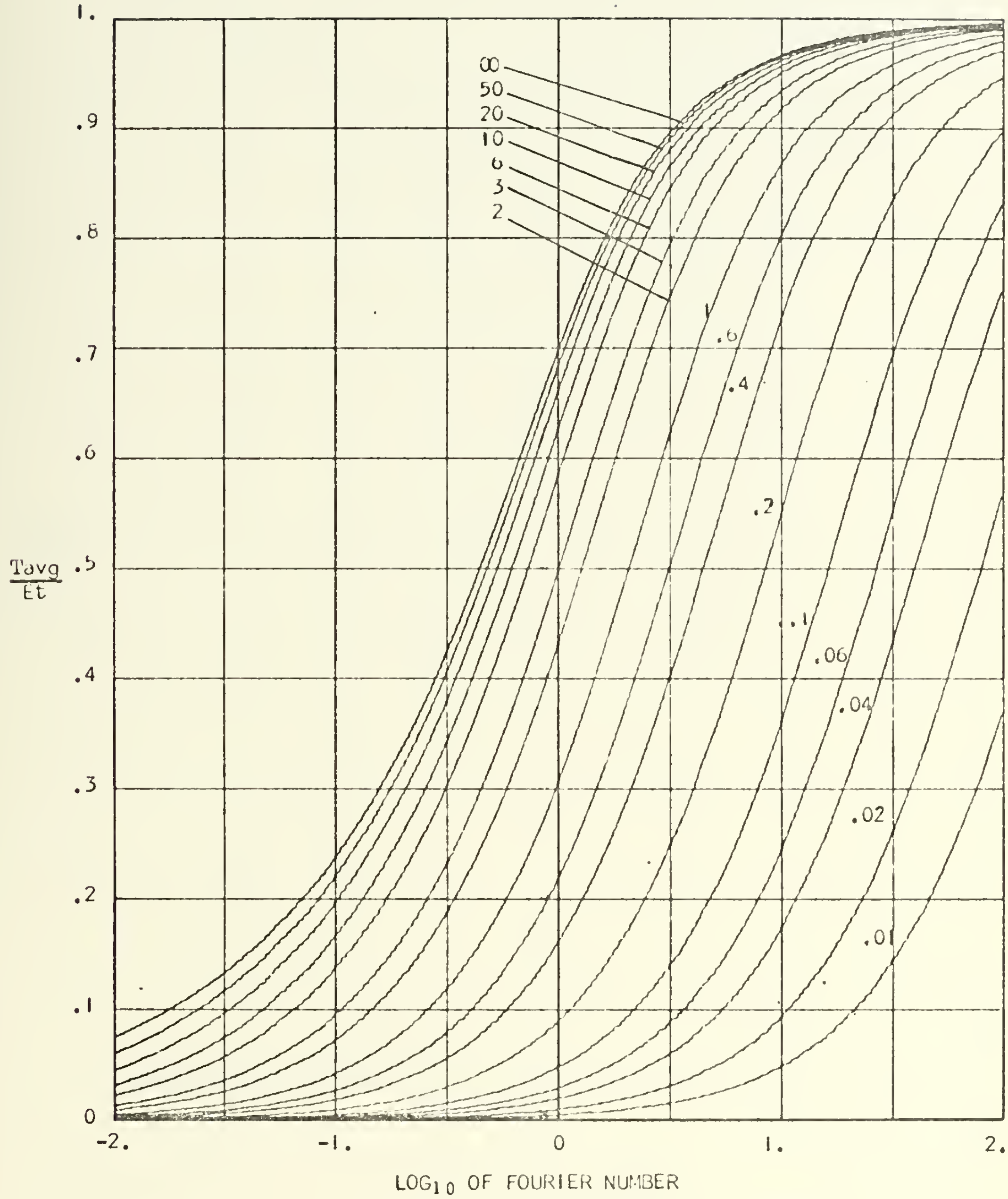
PLOT OF $\Delta T_1/F$ VS. LOG$_{10}$ OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-7



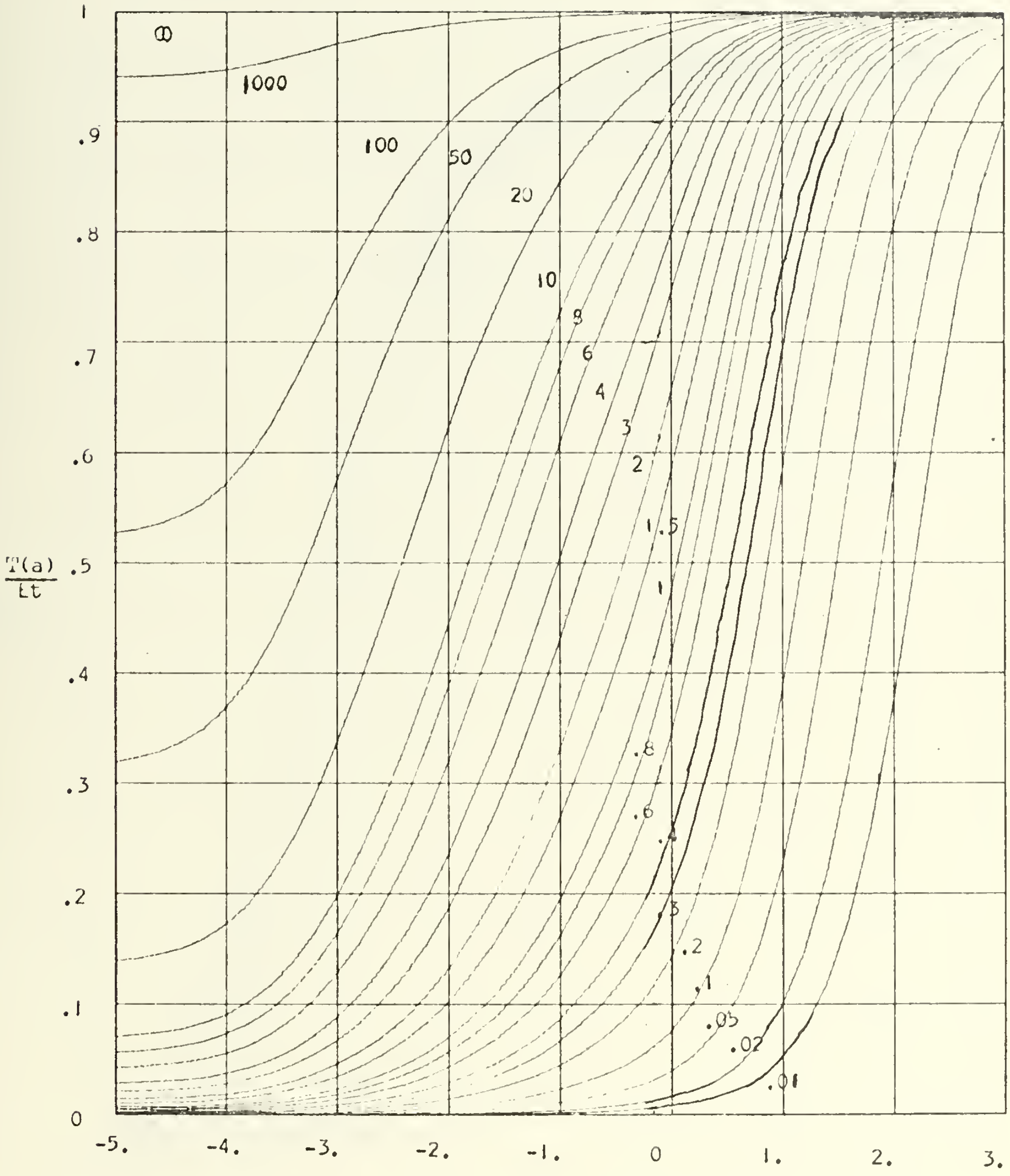
PLOT OF $\Delta T_2/F$ VS. LOG_{10} OF FOURIER NUMBER FOR STEP TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-8



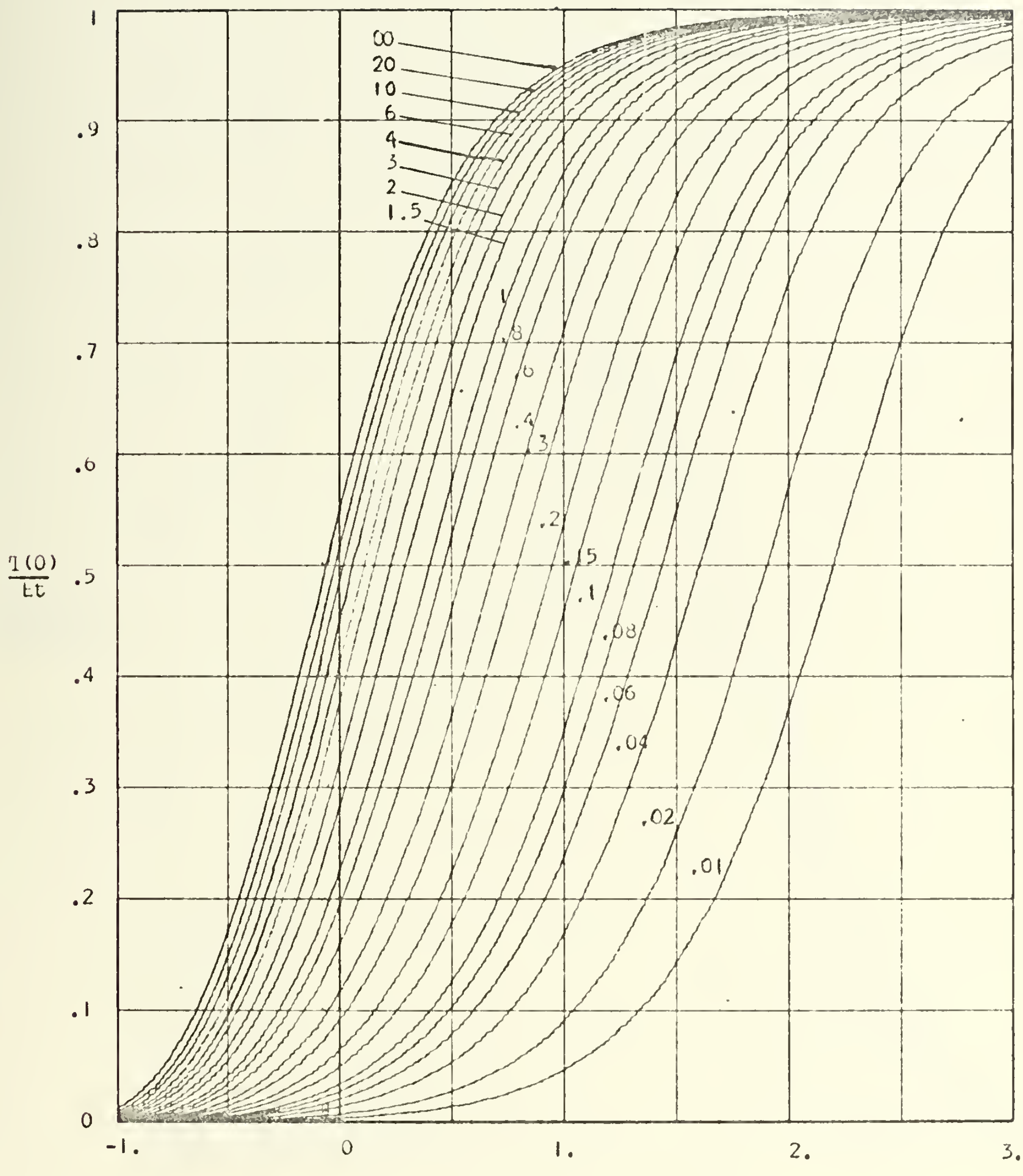
PLOT OF Tavg/Et VS. LOG$_{10}$ OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-9



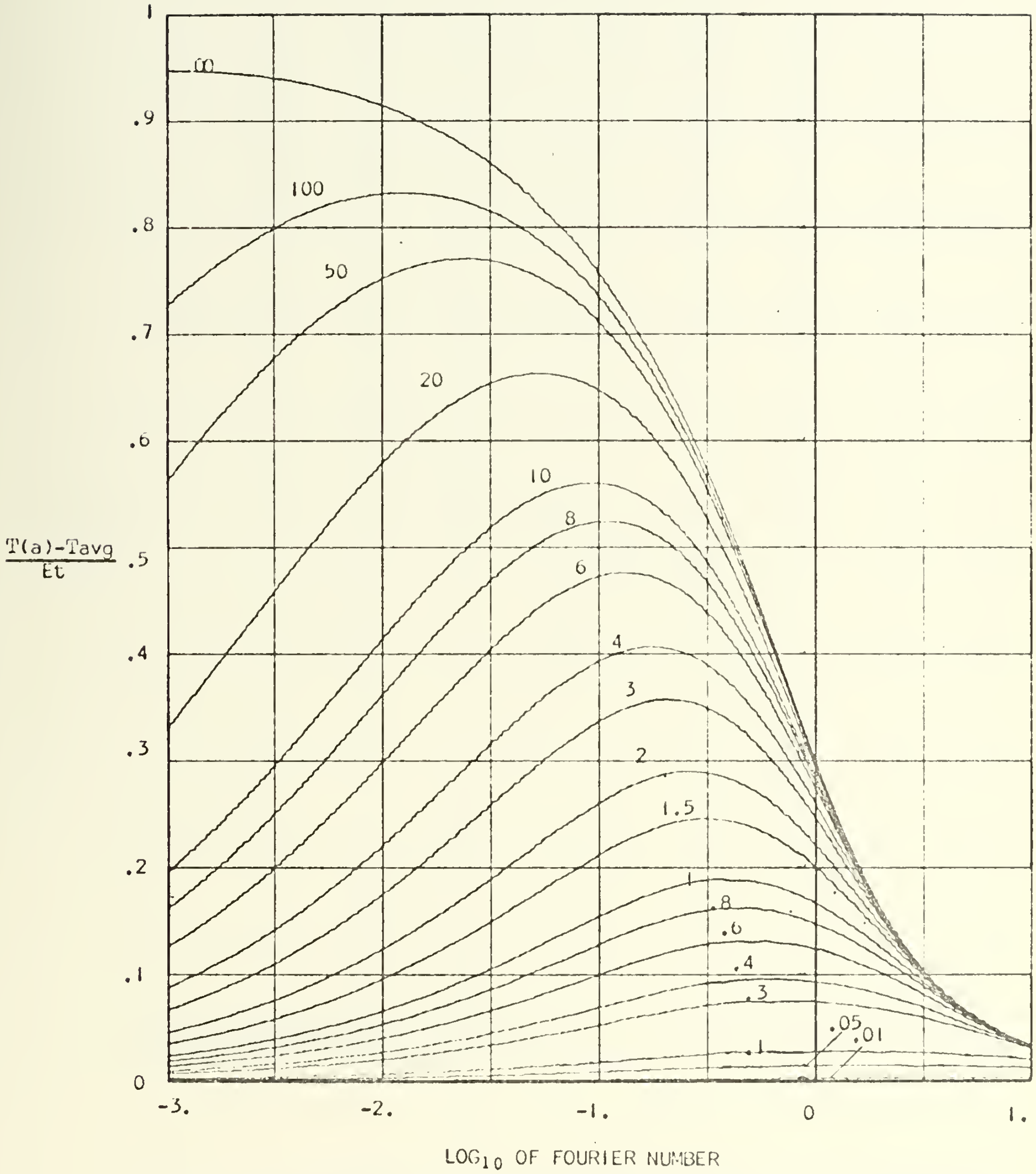
PLOT OF T(a)/Et VS. LOG_{10} OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-10



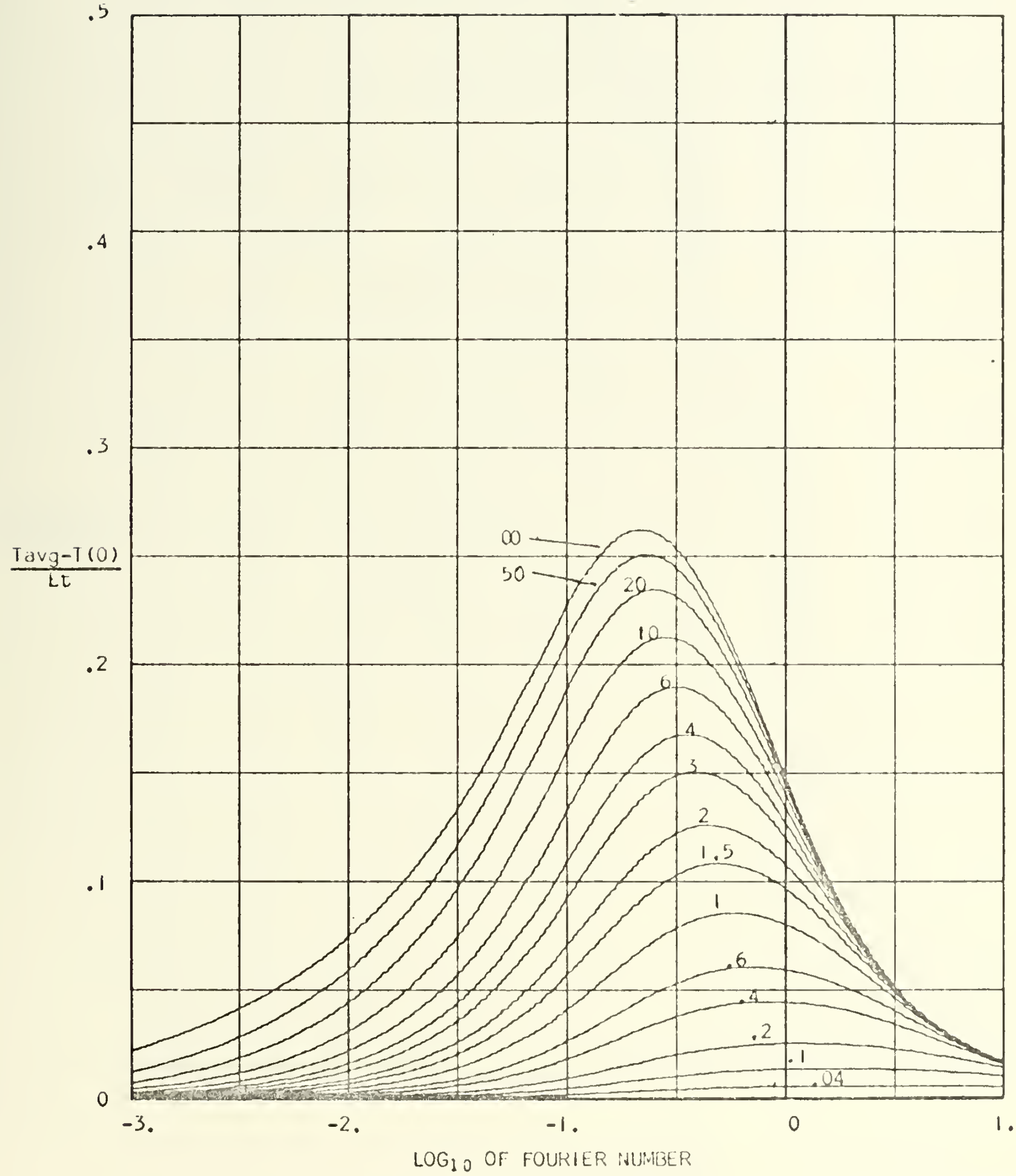
PLOT OF T(0)/Et VS. LOG_{10} OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-11



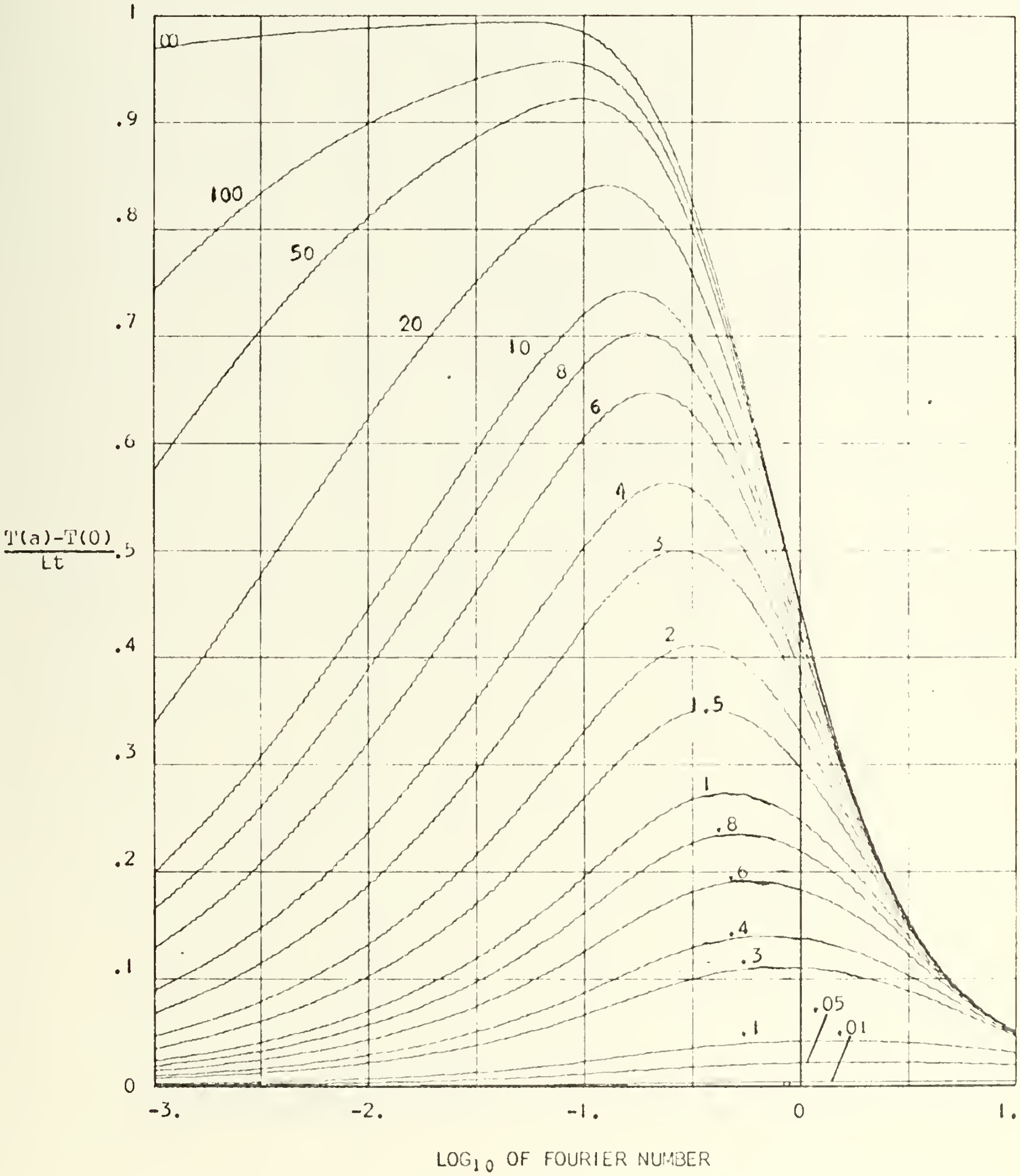
PLOT OF T(a)-Tavg/Et VS. LOG_{10} OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-12



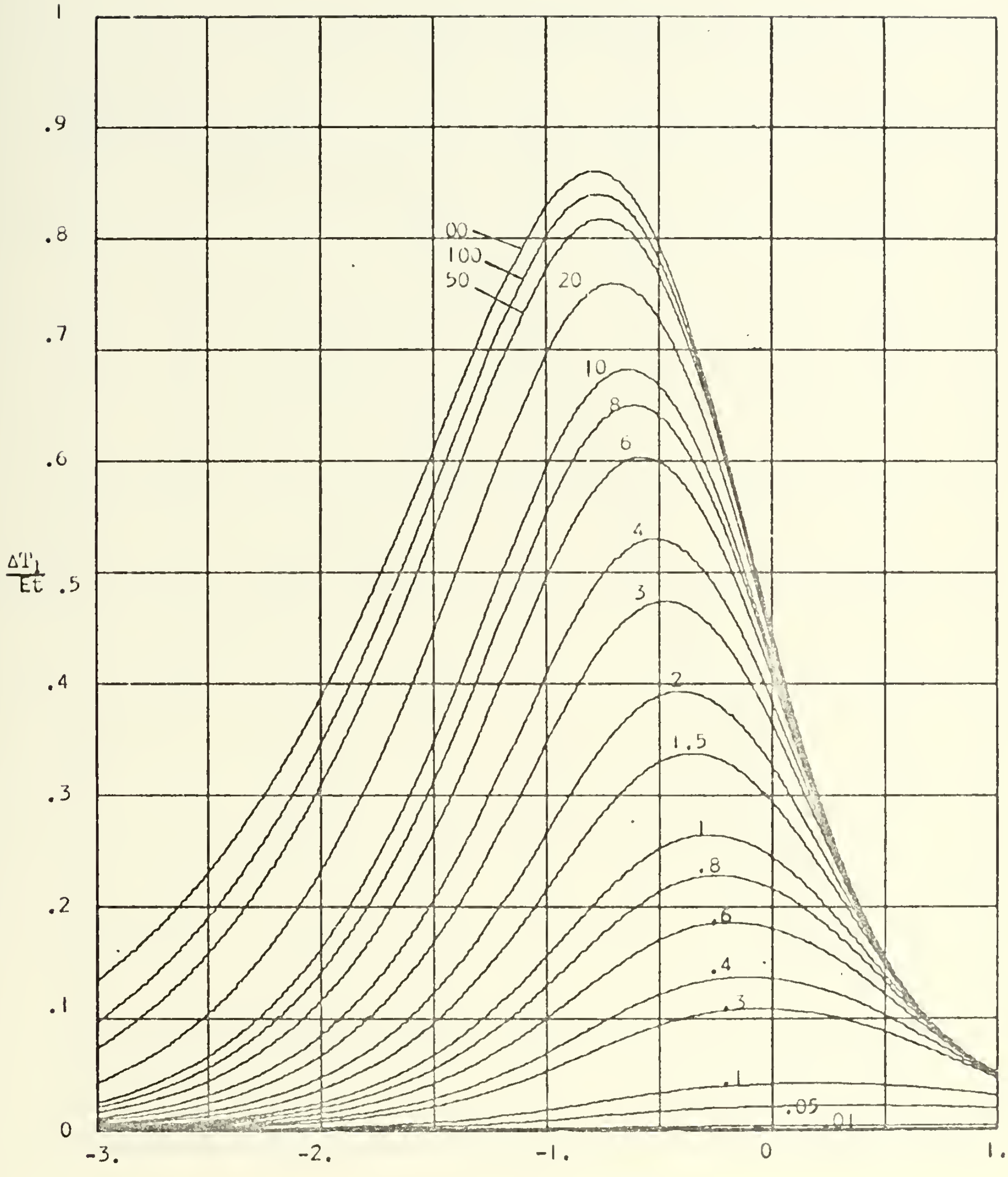
PLOT OF (Tavg - T(0))/Et VS. LOG_{10} OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-13



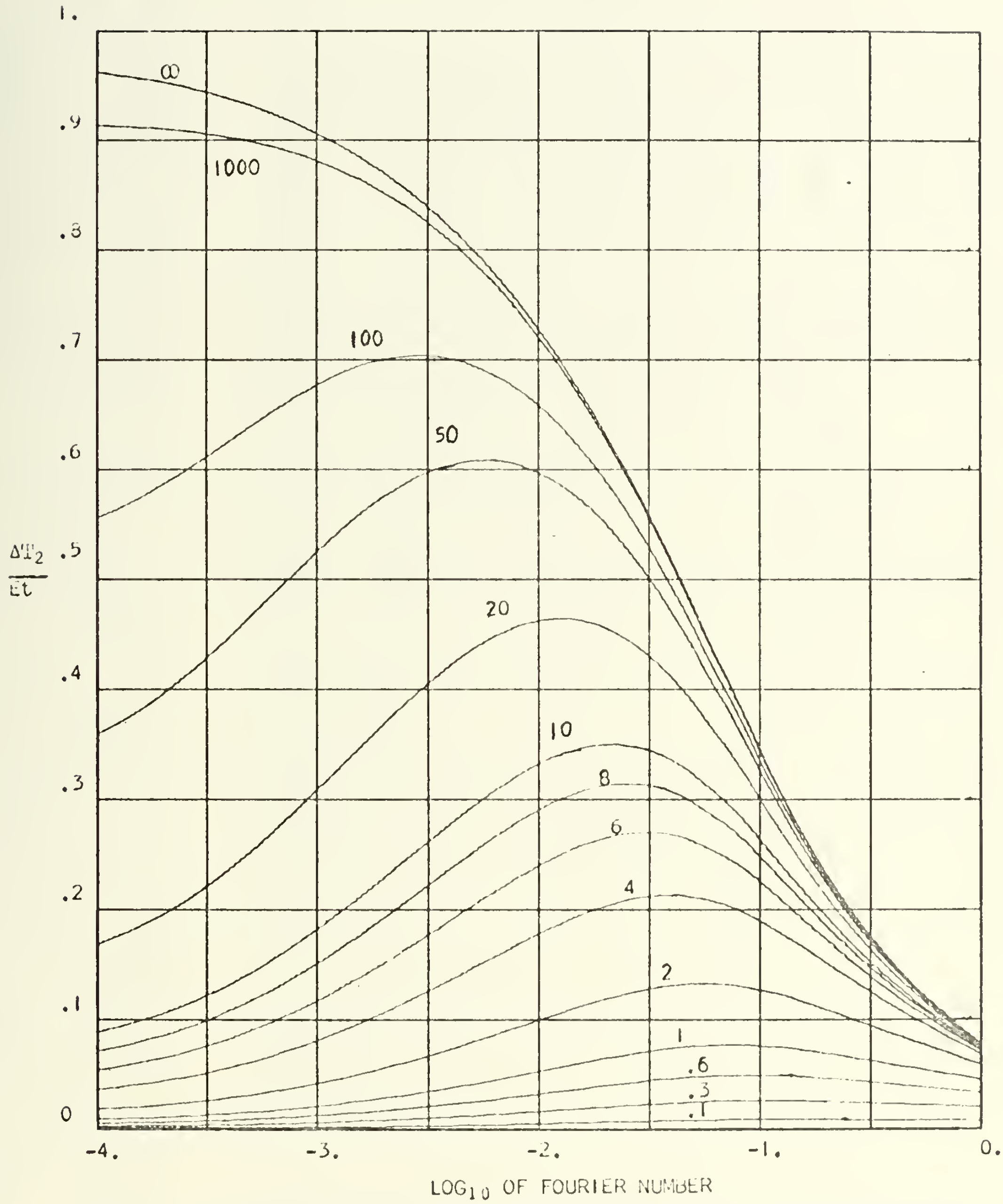
PLOT OF T(a)-T(0)/Et VS. LOG_{10} OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-14



PLOT OF $\Delta T_1/Et$ VS. LOG$_{10}$ OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-15



PLOT OF $\Delta T_2/Et$ VS. LOG_{10} OF FOURIER NUMBER FOR LINEAR TEMPERATURE CHANGE AT VARIOUS BIOT NUMBERS

Fig. A-16

APPENDIX B

COMPUTER PROGRAMS

Section II

```
     THIS PROGRAM IS WRITTEN IN FORTRAN IV FOR THE IBM 360 COMPUTER.  IT SOLVES
THE 16 PROBLEMS DISCUSSED PREVIOUSLY AND PROVIDES FOR GRAPHICAL PRINTOUT OF THE
HEAT TRANSFER CURVES VIA THE CALL DRAW SUBROUTINE AS WELL AS DATA PRINTOUT.  ALL
COMPUTER CARDS, EXCEPT THE COMMENT CARDS, ARE SHOWN AS THEY WOULD BE USED WITH
COLUMNS 1 THROUGH 72 PUNCHED ONLY.  BIOT NUMBER IS CALLED 'NBIOT' AND FOURIER
NUMBER IS CALLED 'NFOURIER'.  'THETA' IS NFOURIER.  'LOGTH' IS LOG TO THE BASE
10 OF NFOURIER.  'CM(K)' AND 'C(J,K)' ARE THE VALUES OF 'C' AS SEEN IN THE FORM-
ULAS.  'EM(K)' AND 'E(J,K)' ARE THE VALUES OF 'M' AS SEEN IN THE FORMULAS.  THE
PROGRAM SOLVES THE EQUATION X*TAN(X)=NBIOT BY THE NEWTON-RAPHSON PROCESS WHERE
X=M AS SEEN IN THE FORMULAS.
```

```
// EXEC FORTCLGP,REGION.FORT=150K,REGION.GO=400K,TIME.GO=15
      IMPLICIT REAL*8 (A-G,L,O-S,U-Z), REAL*4 (H,T), INTEGER (I-K,M,N)
      REAL*8 TITLEA(12)/'D.R.MCNEILL 0814 PLOT OF TAVERAGE DIVIDED BY ET
     1 VS. LOG THETA FOR INFINITE RAMP                          '/
      REAL*8 TITLEB(12)/'D.R.MCNEILL 0814 PLOT OF TAVG MINUS T AT INSULA
     1TED SURFACE DIVIDED BY ET VS. LOG THETA INF. RAMP'/
      REAL*8 TITLEC(12)/'D.R.MCNEILL 0814 PLOT OF T AT HEATED SURFACE MI
     1NUS TAVG DIVIDED BY ET VS. LOG THETA, INF. RAMP   '/
      REAL*8 TITLED(12)/'D.R.MCNEILL 0814 PLOT OF TAVERAGE DIVIDED BY F
     1VS. LOG THETA FOR STEP INPUT                              '/
      REAL*8 TITLEF(12)/'D.R.MCNEILL 0814 PLOT OF TAVG MINUS T AT INSULA
     1TED SURFACE DIVIDED BY F VS. LOG THETA FOR STEP  '/
      REAL*8 TITLEG(12)/'D.R.MCNEILL 0814 PLOT OF T AT HEATED SURFACE MI
     1NUS TAVG DIVIDED BY F VS. LOG THETA FOR STEP      '/
      REAL*8 TITLEH(12)/'D.R.MCNEILL 0814 PLOT OF DELTA T1 DIVIDED BY ET
     1 VS. LOG THETA FOR INFINITE RAMP INPUT                    '/
      REAL*8 TITLEI(12)/'D.R.MCNEILL 0814 PLOT OF DELTA T1 DIVIDED BY F
     1VS. LOG THETA FOR STEP INPUT                              '/
      REAL*8 TITLEJ(12)/'D.R.MCNEILL 0814 PLOT OF DELTA T2 DIVIDED BY F
     1VS. LOG THETA FOR STEP INPUT                              '/
      REAL*8 TITLEK(12)/'D.R.MCNEILL 0814 PLOT OF DELTA T2 DIVIDED BY ET
     1 VS. LOG THETA FOR RAMP INPUT                             '/
      REAL*8 TITLEL(12)/'D.R.MCNEILL 0814 PLOT OF T AT HEATED SURF. MINU
     1S T AT INS. SURF. DIV. BY ET VS. LOG THETA, RAMP '/
      REAL*8 TITLEM(12)/'D.R.MCNEILL 0814 PLOT OF T AT HEATED SURF. MINU
     1S T AT INS. SURF. DIV. BY F VS. LOG THETA, STEP  '/
      REAL*8 TITLEN(12)/'D.R.MCNEILL 0814 PLOT OF T AT INSULATED SURFACE
     1 DIVIDED BY ET VS. LOG THETA FOR RAMP INPUT      '/
      REAL*8 TITLEO(12)/'D.R.MCNEILL 0814 PLOT OF T AT HEATED SURFACE DI
     1VIDED BY ET VS. LOG THETA FOR RAMP INPUT         '/
      REAL*8 TITLEP(12)/'D.R.MCNEILL 0814 PLOT OF T AT INSULATED SURFACE
     1 DIVIDED BY F VS. LOG THETA FOR STEP INPUT       '/
      REAL*8 TITLEQ(12)/'D.R.MCNEILL 0814 PLOT OF T AT HEATED SURFACE DI
     1VIDED BY F VS. LOG THETA FOR STEP INPUT          '/
      REAL LABEL/4H    /
```

```
IN THE DIMENSION STATEMENT, THE FOLLOWING NUMBERS ARE:  23= HIGHEST VALUE OF
'KELL' (THE NUMBER OF NBIOTS) FOUND IN THE PROGRAM; 41 = 'KALL' (NUMBER OF
POINTS PLOTTED FOR EACH NBIOT); AND 30 = NROOT (SEE BELOW).
      DIMENSION BN(23), ARG(41,30), PT(41), T(41), EM(30), CM(30), BM(30
     1), DM(30), C(23,30), E(23,30), B(23,30), D(23,30), LOGTH(41), HOT(
     141), ARGA(41,30), ARGO(41,30), PTB(41)
      F(X,Y)=Y-X*DTAN(X)                                                 2
      G(X)=(X*DTAN(X)+1.)*DTAN(X)+X
      Z=1.                                                              4
      PI=2.*DARSIN(Z)
      Z=1.0D0
      PI=2.0D0*DARSIN(Z)
      PIT=PI/2.0D0
      GAS=174.0D0
'GAS' IS THE MAXIMUM NEGATIVE POWER TO WHICH THE BASE OF THE NATURAL LOGARITHM
MAY BE TAKEN WITHOUT CAUSING COMPUTER UNDERFLOW.
      KALL=41
'KALL' IS THE NUMBER OF 'LOGTH' POINTS CALCULATED.
      FANG=.1
'FANG' IS THE INCREMENT BETWEEN 'LOGTH' VALUES.
      NROOT=30
'NROOT' IS THE NUMBER OF ROOTS CALCULATED OF X*TAN(X)=NBIOT. 30 ROOTS WERE FOUND
TO GIVE VERY PRECISE RESULTS.  10 ROOTS ARE PROBABLY SUFFICIENT.
      EPSIL=1.D-8
'EPSIL' IS THE MAXIMUM DIFFERENCE BETWEEN SUCCESSIVE ROOTS IN CALCULATING THE
ROOTS OF X*TAN(X)=NBIOT.
      KELL=1
'KELL' IS THE NUMBER OF NBIOTS USED FOR THE DIFFERENT FORMULAS.  IT CAN BE SEEN
THAT THE VALUES OF KELL AND OF THE NBIOTS 'BN(J)' ARE CHANGED THROUGHOUT THE
PROGRAM AS REQUIRED BY THE PARTICULAR PROBLEM FOR A GOOD SPACING OF CURVES.
      BN(1)=1.0D0
      JAZZ=1
'JAZZ' IS A CHANGING NUMBER USED TO RETURN TO THE ROOT CALCULATING PORTION FROM
A PARTICULAR PROBLEM.
899   CONTINUE
START OF ROOT CALCULATING PORTION (X*TAN(X)=NBIOT).
      DO 635 J=1,KELL
      Y=BN(J)
      GLUB=.1
4     CONTINUE
      X=PIT-GLUB
      K=1
9     CONTINUE
10    XP=X+F(X,Y)/G(X)
      IF(DABS(XP-X).LT.EPSIL) GO TO 11
      X=XP
      GO TO 10
11    EM(K)=XP
```

```
ROOT 'XP' IS CALCULATED AND STORED AS 'EM(K)' WHERE K IS THE INDEX FOR THE
NUMBER OF ROOTS.
      IF (K.NE.1) GO TO 8
      IF (EM(1).GE.PIT) GLUB=.1*GLUB
      IF (EM(1).GE.PIT) GO TO 4
8     CONTINUE
      IF (K.GE.NROOT) GO TC 12
      X=XP+PI
      K=K+1
      GO TO 9
12    CONTINUE
      DO 13 K=1,NROOT
      A=EM(K)
      CM(K)=4.*DSIN(A)/(2.*A+DSIN(2.*A))
      BM(K)=EM(K)**2
13    DM(K)=EM(K)**3
      IF (EM(1).GT.0.AND.EM(1).LT.PIT) GO TO 15
      PRINT 14,Y
14    FORMAT(5X,'ERRORSTOP, BIOT= ',F20.10)
      GO TO 601
15    CONTINUE
      DO 634 K=1,NROOT
      E(J,K)=EM(K)
      C(J,K)=CM(K)
      B(J,K)=BM(K)
      D(J,K)=DM(K)
634   CONTINUE
635   CONTINUE
END CF ROOT CALCULATING PORTION.
      GO TO(900,901,902,903,904,905,906), JAZZ
900   CONTINUE
      KELL=17
      BN(1)=1000.0D0
      BN(2)=50.0D0
      BN(3)=20.0D0
      BN(4)=10.0D0
      BN(5)=6.0D0
      BN(6)=3.0D0
      BN(7)=2.0D0
      BN(8)=1.0D0
      BN(9)=.6D0
      BN(10)=.4D0
      BN(11)=.2D0
      BN(12)=.1D0
      BN(13)=.06D0
      BN(14)=.04D0
      BN(15)=.02
      BN(16)=.01D0
```

```
      BN(17)=1.0D-8
      JAZZ=2
      GO TO 899
901   CONTINUE
TAVG FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,951)
951   FORMAT (1H1,5X,'TAVG / ET')
      DO 642 J=1,KELL
      DO 6421 I=1,KALL
      LOGTH(I)=-2.+(I-1)*FANG
      HOT(I)=LOGTH(I)
'HOT' IS THE SINGLE PRECISION VALUE OF 'LOGTH' REQUIRED FOR THE X-AXIS ARRAY IN
THE CALL DRAW STATEMENT.
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 801
      ARG(I,1)=  C(J,1)*DSIN(E(J,1))*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(D
     1(J,1)*10.**LOGTH(I))
      GO TO 802
801   ARG(I,1)=  C(J,1)*DSIN(E(J,1))/(D(J,1)*10.**LOGTH(I))
802   CONTINUE
      DO 643 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 803
      ARG(I,K)=ARG(I,K-1)     +C(J,K)*DSIN(E(J,K))*(1.-DEXP(-B(J,K)*10.**
     1LOGTH(I)))/(D(J,K)*10.**LOGTH(I))
      GO TO 804
803   ARG(I,K)=ARG(I,K-1)     +C(J,K)*DSIN(E(J,K))/(D(J,K)*10.**LOGTH(I))
804   CONTINUE
643   CONTINUE
      PT(I)=1.-ARG(I,NROOT)
      T(I)=PT(I)
'T' IS THE SINGLE PRECISION VALUE OF 'PT' (NON-DIMENSIONAL TEMPERATURE) REQUIRED
FOR THE Y-AXIS ARRAY OF THE CALL DRAW STATEMENT.
6421  CONTINUE
      WRITE(6,102) J,(T(I), I=1,KALL)
102   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEA,0,0,0,0,0,0,8,10,1,LAST)
642   CONTINUE
TAVG FOR STEP TEMPERATURE CHANGE
      WRITE(6,954)
954   FORMAT (1H1,5X,'TAVG/F')
      DO 648 J=1,KELL
      DO 6481 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 813
      ARG(I,1)=  C(J,1)/E(J,1)*DSIN(E(J,1))*DEXP(-B(J,1)*10.**LOGTH(I))
```

```
      GO TO 814
813   ARG(I,1)=0.
814   CONTINUE
      DO 649 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 815
      ARG(I,K)=ARG(I,K-1)     +C(J,K)/E(J,K)*DSIN(E(J,K))*DEXP(-B(J,K)*10
     1.**LOGTH(I))
      GO TO 816
815   ARG(I,K)=ARG(I,K-1)
816   CONTINUE
649   CONTINUE
      PT(I)=1.-ARG(I,NROOT)
      T(I)=PT(I)
6481  CONTINUE
      WRITE(6,105) J,(T(I), I=1,KALL)
105   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLED,0,0,0,0,0,0,8,10,1,LAST)
648   CONTINUE
      KELL=15
      BN(1)=1000.0D0
      BN(2)=50.0D0
      BN(3)=20.0D0
      BN(4)=10.0D0
      BN(5)=6.0D0
      BN(6)=4.0D0
      BN(7)=3.0D0
      BN(8)=2.0D0
      BN(9)=1.5D0
      BN(10)=1.0D0
      BN(11)=.6D0
      BN(12)=.4D0
      BN(13)=.2D0
      BN(14)=.1D0
      BN(15)=.04D0
      JAZZ=3
      GO TO 899
902   CONTINUE
TAVG - T(0) FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,952)
952   FORMAT (1H1,5X,'TAVG-T(0)/ET')
      DO 644 J=1,KELL
      DO 6441 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 805
```

```
      ARG(I,1)=  C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(B(J,1)*10.**LO
     1GTH(I))*(1.-DSIN(E(J,1))/E(J,1))
      GO TO 806
805   ARG(I,1)=  C(J,1)*(1.-DSIN(E(J,1))/E(J,1))/(B(J,1)*10.**LOGTH(I))
806   CONTINUE
      DO 645 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 807
      ARG(I,K)=ARG(I,K-1)     +C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/(B
     1(J,K)*10.**LOGTH(I))*(1.-DSIN(E(J,K))/E(J,K))
      GO TO 808
807   ARG(I,K)=ARG(I,K-1)     +C(J,K)*(1.-DSIN(E(J,K))/E(J,K))/(B(J,K)*10
     1.**LOGTH(I))
808   CONTINUE
645   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6441  CONTINUE
      WRITE(6,103) J,(T(I), I=1,KALL)
103   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEB,0,0,0,0,0,0,8,10,1,LAST)
644   CONTINUE
TAVG - T(O) FOR STEP TEMPERATURE CHANGE
      WRITE(6,955)
955   FORMAT (1H1,5X,'TAVG-T(O)/F')
      DO 650 J=1,KELL
      DO 6501 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 817
      ARG(I,1)=  C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))*(1.-DSIN(E(J,1))/E(J
     1,1))
      GO TO 818
817   ARG(I,1)=0.
818   CONTINUE
      DO 651 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 819
      ARG(I,K)=ARG(I,K-1)     +C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))*(1.-DSI
     1N(E(J,K))/E(J,K))
      GO TO 820
819   ARG(I,K)=ARG(I,K-1)
820   CONTINUE
651   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6501  CONTINUE
```

```
      WRITE(6,106) J,(T(I), I=1,KALL)
106   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEF,0,0,0,0,0,0,8,10,1,LAST)
650   CONTINUE
      KELL=19
      BN(1)=1000.0D0
      BN(2)=100.0D0
      BN(3)=50.0D0
      BN(4)=20.0D0
      BN(5)=10.0D0
      BN(6)=8.0D0
      BN(7)=6.0D0
      BN(8)=4.0D0
      BN(9)=3.0D0
      BN(10)=2.0D0
      BN(11)=1.5D0
      BN(12)=1.0D0
      BN(13)=.8D0
      BN(14)=.6D0
      BN(15)=.4D0
      BN(16)=.3D0
      BN(17)=.1D0
      BN(18)=.05D0
      BN(19)=.01D0
      JAZZ=4
      GO TO 899
903   CONTINUE
T(A) - TAVG FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,953)
953   FORMAT (1H1,5X,'T(A)-TAVG/ET')
      DO 646 J=1,KELL
      DO 6461 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 809
      ARG(I,1)=  C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(B(J,1)*10.**LO
     1GTH(I))*(DSIN(E(J,1))/E(J,1)-DCOS(E(J,1)))
      GO TO 810
809   ARG(I,1)=  C(J,1)/(B(J,1)*10.**LOGTH(I))*(DSIN(E(J,1))/E(J,1)-DCOS
     1(E(J,1)))
810   CONTINUE
      DO 647 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 811
      ARG(I,K)=ARG(I,K-1)    +C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/(B
     1(J,K)*10.**LOGTH(I))*(DSIN(E(J,K))/E(J,K)-DCOS(E(J,K)))
```

```
      GO TO 812
811   ARG(I,K)=ARG(I,K-1)     +C(J,K)/(B(J,K)*10.**LOGTH(I))*(DSIN(E(J,K))
     1)/E(J,K)-DCOS(E(J,K)))
812   CONTINUE
647   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6461  CONTINUE
      WRITE(6,104) J,(T(I), I=1,KALL)
104   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEC,0,0,0,0,0,0,8,10,1,LAST)
646   CONTINUE
T(A) - TAVG FOR STEP TEMPERATURE CHANGE
      WRITE(6,956)
956   FORMAT (1H1,5X,'T(A)-TAVG/F')
      DO 652 J=1,KELL
      DO 6521 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 821
      ARG(I,1)=  C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))*(DSIN(E(J,1))/E(J,1)
     1-DCOS(E(J,1)))
      GO TO 822
821   ARG(I,1)=0.
822   CONTINUE
      DO 653 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 823
      ARG(I,K)=ARG(I,K-1)     +C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))*(DSIN(E
     1(J,K))/E(J,K)-DCOS(E(J,K)))
      GO TO 824
823   ARG(I,K)=ARG(I,K-1)
824   CONTINUE
653   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6521  CONTINUE
      WRITE(6,107) J,(T(I), I=1,KALL)
107   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEG,0,0,0,0,0,0,8,10,1,LAST)
652   CONTINUE
DELTA T1 FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,957)
```

```
957   FORMAT (1H1,5X,'DELTA T1/ET')
      DO 654 J=1,KELL
      DO 6541 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 825
      ARG(I,1)=  C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/((E(J,1)**4)*10
     1.**LOGTH(I))*(12.*(1.-DCOS(E(J,1)))-6.*E(J,1)*DSIN(E(J,1)))
      GO TO 826
825   ARG(I,1)=  C(J,1)/((E(J,1)**4)*10.**LOGTH(I))*(12.*(1.-DCOS(E(J,1)
     1))-6.*E(J,1)*DSIN(E(J,1)))
826   CONTINUE
      DO 655 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 827
      ARG(I,K)=ARG(I,K-1)     +C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/((
     1E(J,K)**4)*10.**LOGTH(I))*(12.*(1.-DCOS(E(J,K)))-6.*E(J,K)*DSIN(E(
     1J,K)))
      GO TO 828
827   ARG(I,K)=ARG(I,K-1)     +C(J,K)/((E(J,K)**4)*10.**LOGTH(I))*(12.*(1
     1.-DCOS(E(J,K)))-6.*E(J,K)*DSIN(E(J,K)))
828   CONTINUE
655   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6541  CONTINUE
      WRITE(6,108) J,(T(I), I=1,KALL)
108   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEH,0,0,0,0,0,0,8,10,1,LAST)
654   CONTINUE
DELTA T1 FOR STEP TEMPERATURE CHANGE
      WRITE(6,958)
958   FORMAT (1H1,5X,'DELTA T1/F')
      DO 656 J=1,KELL
      DO 6561 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 829
      ARG(I,1)=  C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))/B(J,1)*(12.*(1.-DCOS
     1(E(J,1)))-6.*E(J,1)*DSIN(E(J,1)))
      GO TO 830
829   ARG(I,1)=0.
830   CONTINUE
      DO 657 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 831
      ARG(I,K)=ARG(I,K-1)     +C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))/B(J,K)*
```

```
     1(12.*(1.-DCOS(E(J,K)))-6.*E(J,K)*DSIN(E(J,K)))
      GO TO 832
831   ARG(I,K)=ARG(I,K-1)
832   CONTINUE
657   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6561  CONTINUE
      WRITE(6,109) J,(T(I), I=1,KALL)
109   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEI,0,0,0,0,0,0,8,10,1,LAST)
656   CONTINUE
T(A) - T(O) FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,961)
961   FORMAT (1H1,5X,'T(A)-T(O)/ET')
      DO 662 J=1,KELL
      DO 6621 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 841
      ARG(I,1)=C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(B(J,1)*10.**LOGT
     1H(I))*(1.-DCOS(E(J,1)))
      GO TO 842
841   ARG(I,1)=C(J,1)/(B(J,1)*10.**LOGTH(I))*(1.-DCOS(E(J,1)))
842   CONTINUE
      DO 663 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 843
      ARG(I,K)=ARG(I,K-1)+C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/(B(J,K
     1)*10.**LOGTH(I))*(1.-DCOS(E(J,K)))
      GO TO 844
843   ARG(I,K)=ARG(I,K-1)+C(J,K)/(B(J,K)*10.**LOGTH(I))*(1.-DCOS(E(J,K))
     1)
844   CONTINUE
663   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6621  CONTINUE
      WRITE(6,112) J,(T(I), I=1,KALL)
112   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEL,0,0,0,0,0,0,8,10,1,LAST)
662   CONTINUE
T(A) - T(O) FOR STEP TEMPERATURE CHANGE
```

```
      WRITE(6,962)
962   FORMAT (1H1,5X,'T(A)-T(O)/F')
      DO 664 J=1,KELL
      DO 6641 I=1,KALL
      LOGTH(I)=-3.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 845
      ARG(I,1)=C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))*(1.-DCOS(E(J,1)))
      GO TO 846
845   ARG(I,1)=0.
846   CONTINUE
      DO 665 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 847
      ARG(I,K)=ARG(I,K-1)+C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))*(1.-DCOS(E(
     1J,K)))
      GO TO 848
847   ARG(I,K)=ARG(I,K-1)
848   CONTINUE
665   CONTINUE
      PT(I)=ARG(I,NROOT)
      T(I)=PT(I)
6641  CONTINUE
      WRITE(6,113) J,(T(I), I=1,KALL)
113   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEM,0,0,0,0,0,0,8,10,1,LAST)
664   CONTINUE
      KELL=14
      BN(1)=1.0D+6
      BN(2)=1000.0D0
      BN(3)=100.0D0
      BN(4)=50.0D0
      BN(5)=20.0D0
      BN(6)=10.0D0
      BN(7)=8.0D0
      BN(8)=6.0D0
      BN(9)=4.0D0
      BN(10)=2.0D0
      BN(11)=1.0D0
      BN(12)=.6D0
      BN(13)=.3D0
      BN(14)=.1D0
      JAZZ=5
      GO TO 899
904   CONTINUE
DELTA T2 FOR STEP TEMPERATURE CHANGE
```

```
      WRITE(6,959)
959   FORMAT (1H1,5X,'DELTA T2/F')
      PAL=0.
      DO 658 J=1,KELL
      DO 6581 I=1,KALL
      LOGTH(I)=-4.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 833
      ARG(I,1)=C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))/B(J,1)*(12.*(1.-DCOS(E
     1(J,1)))-6.*E(J,1)*DSIN(E(J,1)))
      ARGA(I,1)=C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))*(DSIN(E(J,1))/E(J,1)-
     1DCOS(E(J,1)))
      ARGO(I,1)=C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))*(1.-DSIN(E(J,1))/E(J,
     11))
      GO TO 834
833   ARG(I,1)=0.
      ARGA(I,1)=0.
      ARGO(I,1)=0.
834   CONTINUE
      DO 659 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 835
      ARG(I,K)=ARG(I,K-1)+C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))/B(J,K)*(12.
     1*(1.-DCOS(E(J,K)))-6.*E(J,K)*DSIN(E(J,K)))
      ARGA(I,K)=ARGA(I,K-1)+C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))*(DSIN(E(J
     1,K))/E(J,K)-DCOS(E(J,K)))
      ARGO(I,K)=ARGO(I,K-1)+C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))*(1.-DSIN(
     1E(J,K))/E(J,K))
      GO TO 836
835   ARG(I,K)=ARG(I,K-1)
      ARGA(I,K)=ARGA(I,K-1)
      ARGO(I,K)=ARGO(I,K-1)
836   CONTINUE
659   CONTINUE
      PT(I)=DABS(ARGA(I,NROOT))-DABS(ARG(I,NROOT))/2.
      PTB(I)=DABS(ARGO(I,NROOT))-DABS(ARG(I,NROOT))/2.
      T(I)=DMAX1(PT(I),PTB(I),PAL)
6581  CONTINUE
      WRITE(6,110) J,(T(I), I=1,KALL)
110   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEJ,0,0,0,0,0,0,8,10,1,LAST)
658   CONTINUE
DELTA T2 FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,960)
960   FORMAT (1H1,5X,'DELTA T2/ET')
      DO 660 J=1,KELL
```

```
      DO 6601 I=1,KALL
      LOGTH(I)=-4.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 837
      ARG(I,1)=C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/((E(J,1)**4)*10.*
     1*LOGTH(I))*(12.*(1.-DCOS(E(J,1)))-6.*E(J,1)*DSIN(E(J,1)))
      ARGA(I,1)=C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(B(J,1)*10.**LOGTH(I))*(
     1TH(I))*(DSIN(E(J,1))/E(J,1)-DCOS(E(J,1)))
      ARGO(I,1)=C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(B(J,1)*10.**LOG
     1TH(I))*(1.-DSIN(E(J,1))/E(J,1))
      GO TO 838
837   ARG(I,1)=C(J,1)/((E(J,1)**4)*10.**LOGTH(I))*(12.*(1.-DCOS(E(J,1)))
     1-6.*E(J,1)*DSIN(E(J,1)))
      ARGA(I,1)=C(J,1)/(B(J,1)*10.**LOGTH(I))*(DSIN(E(J,1))/E(J,1)-DCOS(
     1E(J,1)))
      ARGO(I,1)=C(J,1)/(B(J,1)*10.**LOGTH(I))*(1.-DSIN(E(J,1))/E(J,1))
838   CONTINUE
      DO 661 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 839
      ARG(I,K)=ARG(I,K-1)+C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/((E(J,
     1K)**4)*10.**LOGTH(I))*(12.*(1.-DCOS(E(J,K)))-6.*E(J,K)*DSIN(E(J,K)
     1))
      ARGA(I,K)=ARGA(I,K-1)+C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/(B(J,K)
     1,K)*10.**LOGTH(I))*(DSIN(E(J,K))/E(J,K)-DCOS(E(J,K)))
      ARGO(I,K)=ARGO(I,K-1)+C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/(B(J
     1,K)*10.**LOGTH(I))*(1.-DSIN(E(J,K))/E(J,K))
      GO TO 840
839   ARG(I,K)=ARG(I,K-1)+C(J,K)/((E(J,K)**4)*10.**LOGTH(I))*(12.*(1.-DC
     1OS(E(J,K)))-6.*E(J,K)*DSIN(E(J,K)))
      ARGA(I,K)=ARGA(I,K-1)+C(J,K)/(B(J,K)*10.**LOGTH(I))*(DSIN(E(J,K))/
     1E(J,K)-DCOS(E(J,K)))
      ARGO(I,K)=ARGO(I,K-1)+C(J,K)/(B(J,K)*10.**LOGTH(I))*(1.-DSIN(E(J,K))
     1))/E(J,K))
840   CONTINUE
661   CONTINUE
      PT(I)=DABS(ARGA(I,NROOT))-DABS(ARG(I,NROOT))/2.
      PTB(I)=DABS(ARGO(I,NROOT))-DABS(ARG(I,NROOT))/2.
      T(I)=DMAX1(PT(I),PTB(I),PAL)
6601  CONTINUE
      WRITE(6,111) J,(T(I), I=1,KALL)
111   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEK,0,0,0,0,0,0,8,10,1,LAST)
660   CONTINUE
      KELL=22
      BN(1)=1000.0D0
```

```
      BN(2)=20.0D0
      BN(3)=10.0D0
      BN(4)=6.0D0
      BN(5)=4.0D0
      BN(6)=3.0D0
      BN(7)=2.0D0
      BN(8)=1.50D0
      BN(9)=1.0D0
      BN(10)=.8D0
      BN(11)=.6D0
      BN(12)=.4D0
      BN(13)=.3D0
      BN(14)=.2D0
      BN(15)=.15D0
      BN(16)=.1D0
      BN(17)=.08D0
      BN(18)=.06D0
      BN(19)=.04D0
      BN(20)=.02D0
      BN(21)=.01D0
      BN(22)=1.0D-8
      JAZZ=6
      GO TO 899
905   CONTINUE
T(0) FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,963)
963   FORMAT (1H1,5X,'T(O)/ET')
      DO 666 J=1,KELL
      DO 6661 I=1,KALL
      LOGTH(I)=-1.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 849
      ARG(I,1)=C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))/(B(J,1)*10.**LOGT
     1H(I))
      GO TO 850
849   ARG(I,1)=C(J,1)/(B(J,1)*10.**LOGTH(I))
850   CONTINUE
      DO 667 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 851
      ARG(I,K)=ARG(I,K-1)+C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))/(B(J,K
     1)*10.**LOGTH(I))
      GO TO 852
851   ARG(I,K)=ARG(I,K-1)+C(J,K)/(B(J,K)*10.**LOGTH(I))
852   CONTINUE
667   CONTINUE
      PT(I)=1.-ARG(I,NROOT)
      T(I)=PT(I)
6661  CONTINUE
```

```
      WRITE(6,114) J,(T(I), I=1,KALL)
114   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEN,0,0,0,0,0,0,8,10,1,LAST)
666   CONTINUE
T(0) FOR STEP TEMPERATURE CHANGE
      WRITE(6,965)
965   FORMAT (1H1,5X,'T(0)/F')
      DO 670 J=1,KELL
      DO 6701 I=1,KALL
      LOGTH(I)=-1.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 857
      ARG(I,1)=C(J,1)*DEXP(-B(J,1)*10.**LOGTH(I))
      GO TO 858
857   ARG(I,1)=0.
858   CONTINUE
      DO 671 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 859
      ARG(I,K)=ARG(I,K-1)+C(J,K)*DEXP(-B(J,K)*10.**LOGTH(I))
      GO TO 860
859   ARG(I,K)=ARG(I,K-1)
860   CONTINUE
671   CONTINUE
      PT(I)=1.-ARG(I,NROOT)
      T(I)=PT(I)
6701  CONTINUE
      WRITE(6,116) J,(T(I), I=1,KALL)
116   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEP,0,0,0,0,0,0,8,10,1,LAST)
670   CONTINUE
      KELL=23
      BN(1)=1000000000.0D0
      BN(2)=1000.0D0
      BN(3)=100.0D0
      BN(4)=50.0D0
      BN(5)=20.0D0
      BN(6)=10.0D0
      BN(7)=8.0D0
      BN(8)=6.0D0
      BN(9)=4.0D0
      BN(10)=3.0D0
      BN(11)=2.0D0
```

```
      BN(12)=1.5D0
      BN(13)=1.0D0
      BN(14)=.80D0
      BN(15)=.60D0
      BN(16)=.40D0
      BN(17)=.30D0
      BN(18)=.20D0
      BN(19)=.1D0
      BN(20)=.05D0
      BN(21)=.02D0
      BN(22)=.01D0
      BN(23)=1.0D-8
      JAZZ=7
      GO TO 899
906   CONTINUE
T(A) FOR LINEAR TEMPERATURE CHANGE
      WRITE(6,964)
964   FORMAT (1H1,5X,'T(A)/ET')
      FANG=.2
      DO 668 J=1,KELL
      DO 6681 I=1,KALL
      LOGTH(I)=-5.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 853
      ARG(I,1)=C(J,1)*(1.-DEXP(-B(J,1)*10.**LOGTH(I)))*DCOS(E(J,1))/(B(J
     1,1)*10.**LOGTH(I))
      GO TO 854
853   ARG(I,1)=C(J,1)*DCOS(E(J,1))/(B(J,1)*10.**LOGTH(I))
854   CONTINUE
      DO 669 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 855
      ARG(I,K)=ARG(I,K-1)+C(J,K)*(1.-DEXP(-B(J,K)*10.**LOGTH(I)))*DCOS(E
     1(J,K))/(B(J,K)*10.**LOGTH(I))
      GO TO 856
855   ARG(I,K)=ARG(I,K-1)+C(J,K)*DCOS(E(J,K))/(B(J,K)*10.**LOGTH(I))
856   CONTINUE
669   CONTINUE
      PT(I)=1.-ARG(I,NROOT)
      T(I)=PT(I)
6681  CONTINUE
      WRITE(6,115) J,(T(I), I=1,KALL)
115   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLE0,0,0,0,0,0,0,8,10,1,LAST)
668   CONTINUE
T(A) FOR STEP TEMPERATURE CHANGE
```

```
      WRITE(6,966)
966   FORMAT (1H1,5X,'T(A)/F')
      DO 672 J=1,KELL
      DO 6721 I=1,KALL
      LOGTH(I)=-5.+(I-1)*FANG
      HOT(I)=LOGTH(I)
      IF((B(J,1)*10.**LOGTH(I)).GT.GAS) GO TO 861
      ARG(I,1)=C(J,1)*DCOS(E(J,1))*DEXP(-B(J,1)*10.**LOGTH(I))
      GO TO 862
861   ARG(I,1)=0.
862   CONTINUE
      DO 673 K=2,NROOT
      IF((B(J,K)*10.**LOGTH(I)).GT.GAS) GO TO 863
      ARG(I,K)=ARG(I,K-1)+C(J,K)*DCOS(E(J,K))*DEXP(-B(J,K)*10.**LOGTH(I)
     1)
      GO TO 864
863   ARG(I,K)=ARG(I,K-1)
864   CONTINUE
673   CONTINUE
      PT(I)=1.-ARG(I,NROOT)
      T(I)=PT(I)
6721  CONTINUE
      WRITE(6,117) J,(T(I), I=1,KALL)
117   FORMAT (5X,'T(I) FOR J=' I3,/,(3F20.10))
      KULL=2
      IF (J.EQ.1) KULL=1
      IF (J.EQ.KELL) KULL=3
      CALL DRAW (KALL,HOT,T,KULL,0,LABEL,TITLEQ,0,0,0,0,0,0,8,10,1,LAST)
672   CONTINUE
601   CONTINUE
      STOP
      END
```

Section III

Generation of Betas and Gammas

```
                 'GENERATION OF BETAS AND GAMMAS'
     THIS PROGRAM SOLVES FOR THE VALUES OF BETA(J,K,L) AND GAMMA(R,S,T)
BY USING THE EXACT SOLUTION TO GET VALUES OF T(J,I,M,INDEX) WHERE INDEX
IS USED TO NOTE THE DIFFERENT DATA POINTS.  THE FIRST PORTION OF THE
PROGRAM IS NEARLY IDENTICAL TO THAT USED FOR GENERATION OF HEAT TRANSFER
CURVES.  THE PROGRAM IS SHOWN FOR THE RAMP CASE.  SUBSTITUTION CARDS FOR THE
STEP CASE ARE SHOWN AT THE END OF THE PROGRAM.  PROGRAM SIZE IS 520K.
      IMPLICIT REAL*8 (A-G,L,P,R,S,X-Z), REAL*4 (T), INTEGER (H-K,M-O,Q,
     1V-W)
      DIMENSION BN(4), ARG(30), EM(30), CM(30), BM(30), T(4,6,6,144), C(
     14,30), E(4,30), THETA(6), TAMMAX(2,4,4,144), TAMMA(2,4,4), TETAX(3
     1,7,7,144), TETA(3,7,7), B(4,30), BNP(4), THETAX(6)
      Z=1.0D0
      PI=2.0D0*DARSIN(Z)
      PIT=PI/2.0D0
      GAS=174.0D0
      KELL=4
      KALL=6
      FANG=10
KELL, KALL, AND FANG ARE USED THE SAME AS IN THE FIRST PROGRAM.
      KASP=6
'KASP' IS THE NUMBER OF DATA POINTS OF X/A CALCULATED.
      FANGO=.2
'FANGO' IS THE INCREMENT OF X/A VALUES.
      FARO=144
'FARO' IS THE NUMBER OF DATA POINTS USED.
      EPSIL=1.D-8
      NROOT=30
BIOT NUMBERS ARE DECLARED AS BN(J)
      BN(1)=.01D0
      BN(2)=1.0D0
      BN(3)=100.0D0
      BN(4)=1000000.0D0
      DO 635 J=1,KELL
'BNP' IS THE FUNCTION OF 'BN' USED IN THE APPROXIMATION.
      BNP(J)=BN(J)/(BN(J)+1.0D0)
      Y=BN(J)
      GLUB=.1
4     CONTINUE
      X=PIT-GLUB
      K=1
9     CONTINUE
      F=Y-X*DTAN(X)
      G=(X*DTAN(X)+1.)*DTAN(X)+X
      XP=X+F/G
      IF(DABS(XP-X).LT.EPSIL) GO TO 11
      X=XP
      GO TO 9
```

```
11     EM(K)=XP
       IF (K.NE.1) GO TO 8
       IF (EM(1).GE.PIT) GLUB=.1*GLUB
       IF (EM(1).GE.PIT) GO TO 4
8      CONTINUE
       IF (K.GE.NROOT) GO TO 12
       X=XP+PI
       K=K+1
       GO TO 9
12     CONTINUE
       DO 13 K=1,NROOT
       A=EM(K)
       CM(K)=4.*DSIN(A)/(2.*A+DSIN(2.*A))
13     BM(K)=EM(K)**2
       IF (EM(1).GT.0.AND.EM(1).LT.PIT) GO TO 15
       PRINT 14,Y
14     FORMAT(5X,'ERRORSTOP, BIOT= ',F20.10)
       GO TO 601
15     CONTINUE
       DO 634 K=1,NROOT
       E(J,K)=EM(K)
       C(J,K)=CM(K)
       B(J,K)=BM(K)
634    CONTINUE
635    CONTINUE
END OF ROOT CALCULATING PORTION.  START OF CALCULATING T(J,I,M,INDEX).
       INDEX=1
       DO 666 J=1,KELL
       DO 6662 M=1,KASP
       XOVA=0.+(M-1)*FANGO
'XOVA' IS THE VALUE OF X/A.
       DO 6661 I=1,KALL
       THETA(I)=.0001*FANG**I
       THETAX(I)=THETA(I)/(1.0+THETA(I))
'THETAX' IS THE FUNCTION OF 'THETA' USED IN THE APPROXIMATION.
       IF((B(J,1)*THETA(I)).GT.GAS) GO TO 849
       ARG(1)=C(J,1)/(B(J,1)*THETA(I))*DCOS(E(J,1)*XOVA)*(1.-DEXP(-B(J,1)
      1*THETA(I)))
       GO TO 850
849    ARG(1)=C(J,1)/(B(J,1)*THETA(I))*DCOS(E(J,1)*XOVA)
850    CONTINUE
       DO 667 K=2,NROOT
       IF((B(J,K)*THETA(I)).GT.GAS) GO TO 851
       ARG(K)=ARG(K-1)+C(J,K)/(B(J,K)*THETA(I))*DCOS(E(J,K)*XOVA)*(1.-DEX
      1P(-B(J,K)*THETA(I)))
       GO TO 852
851    ARG(K)=ARG(K-1)+C(J,K)/(B(J,K)*THETA(I))*DCOS(E(J,K)*XOVA)
852    CONTINUE
```

```
667    CONTINUE
       T(J,I,M,INDEX)=1.-ARG(NROOT)
       INDEX=INDEX+1
6661   CONTINUE
6662   CONTINUE
666    CONTINUE
START  OF GENERATION OF GAMMA VALUES.
       INDEX=1
       DO 970 O=1,2
       DO 971 Q=1,4
       DO 972 W=1,4
       DO 973 J=1,KELL
       DO 974 M=1,KASP
       XOVA=0.+(M-1)*FANGO
       DO 975 I=1,KALL
       IF (INDEX.NE.1) GO TO 990
       TAMMAX(O,Q,W,1)=T(J,I,M,1)*DCOS(XOVA)**(O-1)*THETAX(I)**(Q-1)*BNP(
      1J)**(W-1)
990    CONTINUE
       IF (INDEX.EQ.1) GO TO 991
       TAMMAX(O,Q,W,INDEX)=TAMMAX(O,Q,W,INDEX-1)+T(J,I,M,INDEX)*DCOS(XOVA
      1)**(O-1)*THETAX(I)**(Q-1)*BNP(J)**(W-1)
991    CONTINUE
       IF (INDEX.NE.FARO) GO TO 992
       TAMMA(O,Q,W)=TAMMAX(O,Q,W,FARO)
'TAMMA' IS GAMMA.
       WRITE (6,90) O,Q,W
90     FORMAT (10X,'TAMMA(O,Q,W) FOR ','O=',I2,'Q=',I2,'W=',I2)
       WRITE (6,91) TAMMA(O,Q,W)
91     FORMAT (5X,F20.5)
       INDEX=1
       GO TO 1000
992    CONTINUE
       INDEX=INDEX+1
1000   CONTINUE
975    CONTINUE
974    CONTINUE
973    CONTINUE
972    CONTINUE
971    CONTINUE
970    CONTINUE
START  OF GENERATION OF BETA VALUES.
       INDEX=1
       DO 980 O=1,3
       DO 981 Q=1,7
       DO 982 W=1,7
       DO 983 J=1,KELL
       DO 984 M=1,KASP
```

```
      XOVA=0.+(M-1)*FANGO
      DO 985 I=1,KALL
      IF (INDEX.NE.1) GO TO 995
      TETAX(O,Q,W,1)=DCOS(XOVA)**(O-1)*THETAX(I)**(Q-1)*BNP(J)**(W-1)
995   CONTINUE
      IF (INDEX.EQ.1) GO TO 996
      TETAX(O,Q,W,INDEX)=TETAX(O,Q,W,INDEX-1)+DCOS(XOVA)**(O-1)*THETAX(I
     1)**(Q-1)*BNP(J)**(W-1)
996   CONTINUE
      IF (INDEX.NE.FARO) GO TO 997
      TETA(O,Q,W)=TETAX(O,Q,W,FARO)
'TETA' IS BETA.
      WRITE (6,92) O,Q,W
92    FORMAT (10X, 'TETA(O,Q,W) FOR ','O=',I2,'Q=',I2,'W=',I2)
      WRITE (6,93) TETA(O,Q,W)
93    FORMAT (5X,F20.5)
      INDEX=1
      GO TO 1001
997   CONTINUE
      INDEX=INDEX+1
1001  CONTINUE
985   CONTINUE
984   CONTINUE
983   CONTINUE
982   CONTINUE
981   CONTINUE
980   CONTINUE
601   CONTINUE
      STOP
      END
THE BELOW CARDS SHOULD BE SUBSTITUTED, WHERE APPROPRIATE, FOR THE STEP CASE.
      ARG(1)=C(J,1)*DCOS(E(J,1)*XOVA)*DEXP(-B(J,1)*THETA(I))
849   ARG(1)=0.
      ARG(K)=ARG(K-1)+C(J,K)*DCOS(E(J,K)*XOVA)*DEXP(-B(J,K)*THETA(I))
851   ARG(K)=ARG(K-1)
```

```
                             'CHECK PROGRAM'
     THIS PROGRAM INCLUDES THE COEFFICIENTS A(M,N,P) FOR THE STEP CASE.
COEFFICIENTS FOR THE RAMP CASE ARE DISPLAYED AT THE END OF THE PROGRAM.
      IMPLICIT REAL*4 (A-H,L,O-Z), INTEGER (I-K,M,N)
      DIMENSION THETA(6), ARG(32), A(2,4,4), BN(6), BNP(6), THETAX(6)
      XOVA=0.
      XOVA=1.
'XOVA' IS THE VALUE OF X/A.  THESE CARDS SHOULD BE SWITCHED AFTER ONE RUN TO
GIVE A PRINTOUT FOR BOTH VALUES.
A(M,N,P) FOR STEP CASE.
      A(1,1,1)=.00802287
      A(1,1,2)=.0111088
      A(1,1,3)=.0145189
      A(1,1,4)=-.00158254
      A(1,2,1)=.0469639
      A(1,2,2)=.0524304
      A(1,2,3)=.0544319
      A(1,2,4)=.0405793
      A(1,3,1)=.637086
      A(1,3,2)=.6541176
      A(1,3,3)=.67496
      A(1,3,4)=.68791
      A(1,4,1)=-.0894503
      A(1,4,2)=.841533
      A(1,4,3)=-2.56277
      A(1,4,4)=-.00241147
      A(2,1,1)=.0134795
      A(2,1,2)=.285058
      A(2,1,3)=-1.17559
      A(2,1,4)=.955424
      A(2,2,1)=-.957273
      A(2,2,2)=5.71995
      A(2,2,3)=-6.90335
      A(2,2,4)=5.15831
      A(2,3,1)=.688737
      A(2,3,2)=2.9818
      A(2,3,3)=-12.1797
      A(2,3,4)=2.38766
      A(2,4,1)=-.0191342
      A(2,4,2)=.102714
      A(2,4,3)=-4.80407
      A(2,4,4)=7.69181
BIOT NUMBERS ARE DECLARED AS BN(J)
      BN(1)=.01
      BN(2)=.1
      BN(3)=1.
      BN(4)=10.
      BN(5)=100.
```

```
      BN(6)=1000000.
      KELL=6
      KALL=6
      FANG=10
KELL, KALL, AND FANG ARE USED THE SAME AS IN THE FIRST PROGRAM.
      FARO=32
'FARO' IS THE NUMBER OF COEFFICIENTS IN THE APPROXIMATION.
      DO 700 J=1,KELL
      BNP(J)=BN(J)/(1.0+BN(J))
'BNP' IS THE FUNCTION OF 'BN' USED IN THE APPROXIMATION.
      WRITE (6,90) J
90    FORMAT (5X,'TEMP FOR J=',I2)
      DO 701 I=1,KALL
      THETA(I)=.0001*FANG**I
      THETAX(I)=THETA(I)/(1.0+THETA(I))
'THETAX' IS THE FUNCTION OF 'THETA' USED IN THE APPROXIMATION.
      INDEX=1
      DO 702 K=1,2
      DO 703 M=1,4
      DO 704 N=1,4
      IF (INDEX.NE.1) GO TO 900
      ARG(1)=A(1,1,1)
      GO TO 901
900   CONTINUE
      ARG(INDEX)=ARG(INDEX-1)+A(K,M,N)*COS(XOVA)**(K-1)*THETAX(I)**(M-1)
     1*BNP(J)**(N-1)
901   CONTINUE
      IF (INDEX.NE.FARO) GO TO 902
      TEMP=ARG(FARO)
'TEMP' IS THE TEMPERATURE SOLVED.  IT IS WRITTEN FOR A GIVEN BIOT NUMBER SUCH
THAT EACH FOURIER NUMBER INCREMENT IS PRINTED.
      WRITE (6,91) TEMP
91    FORMAT (5X,F10.5)
902   CONTINUE
      INDEX=INDEX+1
704   CONTINUE
703   CONTINUE
702   CONTINUE
701   CONTINUE
700   CONTINUE
      STOP
      END
VALUES OF A(M,N,P) FOR THE RAMP CASE ARE LISTED BELOW.
      A(1,1,1)=.0223365
      A(1,1,2)=.011287
      A(1,1,3)=.0169389
      A(1,1,4)=.00381532
      A(1,2,1)=.055696
```

```
A(1,2,2)=.0604526
A(1,2,3)=.0646492
A(1,2,4)=.0482695
A(1,3,1)=.624756
A(1,3,2)=.634413
A(1,3,3)=1.14826
A(1,3,4)=.647658
A(1,4,1)=-.281942
A(1,4,2)=1.10192
A(1,4,3)=-3.0706
A(1,4,4)=.0213157
A(2,1,1)=-.012431
A(2,1,2)=.104461
A(2,1,3)=-.387831
A(2,1,4)=.348023
A(2,2,1)=-.656105
A(2,2,2)=-.485788
A(2,2,3)=1.32394
A(2,2,4)=1.13619
A(2,3,1)=.441186
A(2,3,2)=3.09013
A(2,3,3)=-2.27264
A(2,3,4)=-5.2019
A(2,4,1)=-.0120974
A(2,4,2)=.0994219
A(2,4,3)=-4.84179
A(2,4,4)=7.20499
```

BIBLIOGRAPHY

1. Conduction Heat Transfer, Paul J. Schneider, Addison Wesley Pub. Co. Inc., Cambridge, 1955.

2. Technical Report E-1086, AEC-RDT Piping Design Guide, Teledyne Materials Research Company, Waltham, Mass., June 27, 1969. pp. 64-69.

3. Design Guide for LMFBR Sodium Piping, Volume II Procedures, (Preliminary Draft), prepared for U. S. Atomic Energy Commission by C. F. Brown and Co., Alhambra, Calif., August 22, 1969, pp. 6-10 through 6-13.

4. "Temperature Charts for Induction and Constant-Temperature Heating," by M. P. Heisler, Transactions of the ASME, April, 1947.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 212 Naval Postgraduate School Monterey, California	2
3.	Prof. J. E. Brock, Code 59Bc Mechanical Engineering Department Naval Postgraduate School Monterey, California 93940	
4.	Naval Ship Systems Command Code 2052 Via Code 031 Department of the Navy Washington, D. C. 20360	1
5.	LCDR D. R. McNeill, USN 1454 Hollenbeck Ave., Sunnyvale, California 94085	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, Calif. 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3 REPORT TITLE

Heat Transfer in Infinite Slabs

4 DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1970

5. AUTHOR(S) (First name, middle initial, last name)

Donald Ray McNeill

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1970	72	4

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

An analysis is made of temperature distributions in a slab of finite thickness and infinite extent, one surface of which is perfectly insulated and the other surface of which is exposed to a fluid the temperature of which varies in a specified way. Two cases are considered. In the first case, the temperature of the fluid is suddenly changed and maintained at the new value. In the second case the temperature of the fluid increases linearly with time. From the solutions of these problems, curve sheets are developed which permit evaluating each of eight different quantities of engineering significance. These curve sheets cover more cases of interest and are easier to use than similar curves which have appeared in the report literature.

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Heat conduction						
Temperature distribution						
Pipe wall temperature						

124778
Thesis
M262 McNeill
c.1 Heat transfer in
infinite slabs.